

NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET
FAKULTET FOR FYSIKK , INFORMATIKK OG MATEMATIKK



HOVEDOPPGAVE

Kandidatens navn: Ulf Carlin
Fag: Datateknikk
Oppgavens tittel (norsk):
Oppgavens tittel (engelsk): **Mining medical data with rough sets**
Oppgavens tekst:

Rough sets appear to be a theoretically well-founded approach to data mining that has been made practical since the development of the ROSETTA system. The purpose of this diploma is to investigate the specific requirements for data mining in medical domains using that system. The experimental part shall contain a study of a real world dataset concerned with appendicitis that was previously analysed using logistic regression and subsequently published in the medical community. Specifically, diagnostic rules are to be synthesized and their quality compared with the former result.

Oppgaven gitt: 1. oktober 1997
Besvarelsen leveres innen: 23. februar 1998
Besvarelsen levert: 23. februar 1998
Utført ved: Institutt for datateknikk og informasjonsvitenskap (IDI)
Veileder: Mihhail Matskin

Trondheim, 25. februar 1998

Mihhail Matskin
Faglærer

Mining medical data with rough sets

Ulf Carlin

Knowledge Systems Group
Department of Computer and Information Science
Faculty of Physics, Informatics and Mathematics
The Norwegian University of Science and Technology
N-7034 Trondheim, Norway
e-mail: usc@idi.ntnu.no

Trondheim,
February 23, 1998

Abstract

A data set describing 257 patients with acute appendicitis has been analysed in this thesis using rough set tools. The data set has previously been studied successfully using logistic regression. The current rough set analysis has shown to perform approximately equally well as the logistic regression analysis. The rough set approach is typically said to offer an additional advantage of a «white-box» presentation of rules in the form of simple propositional rules. One problem faced was that the rough set models resulted in rule bases consisting of thousands of rules. This challenge was met by using a number of different «filtering» strategies. The rule bases could then be reduced, often considerably, with only a small decrease in the performance. The model performance was measured using ROC curves, which is a performance measure that should be used more frequently in assessing the performance of data mining methods.

Table of Contents

1. INTRODUCTION	1
1.1 DATA MINING AND KNOWLEDGE DISCOVERY.....	2
1.2 INTENDED READER	2
1.3 THE STRUCTURE OF THE THESIS	2
2. PRELIMINARY NOTIONS: ROUGH SETS	3
2.1 INFORMATION SYSTEMS, DECISION SYSTEMS AND INDISCERNIBILITY	3
2.2 REDUCTS.....	3
2.3 OTHER REDUCT TYPES	4
2.3.1 <i>Object-related Reducts</i>	4
2.3.2 <i>Dynamic Reducts</i>	5
2.4 REDUCTS MODULO THE DECISION ATTRIBUTE.....	6
2.5 DECISION RULES	6
2.6 THE ROSETTA SYSTEM.....	7
3. MODEL EVALUATION	8
3.1 SIMPLE VALIDATION	8
3.2 CROSS-VALIDATION	8
3.3 TEST PERFORMANCE STATISTICS	8
3.3.1 <i>Simple Measures from the Contingency Matrix</i>	9
3.3.2 <i>ROC analysis</i>	10
4. DATA MATERIAL AND INTRODUCTORY ANALYSES	13
4.1 DATA MATERIAL.....	13
4.1.1 <i>Discretization</i>	14
4.2 INTRODUCTORY ANALYSES.....	15
4.2.1 <i>Analysis of Each Binary Variable Used on Its Own</i>	15
4.2.2 <i>Analysis of the Surgeons' Classification</i>	16
5. GENERAL METHODOLOGY	18
6. EXPERIMENTS	21
6.1 EXPERIMENT 1	21
6.1.1 <i>Filtering on rule coverage</i>	21
6.1.2 <i>Filtering on rule probability</i>	22
6.1.3 <i>Filtering on rule stability</i>	23
6.1.4 <i>Filtering on reduct length</i>	24
6.1.5 <i>Filtering on reduct support</i>	24
6.1.6 <i>Filtering by removing single attributes</i>	25
6.1.7 <i>Comparison</i>	25
6.2 EXPERIMENT 2	26
6.2.1 <i>Filtering on rule coverage</i>	27
6.2.2 <i>Filtering on rule probability</i>	29
6.2.3 <i>Filtering on reduct length</i>	29
6.2.4 <i>Filtering on reduct support</i>	30
6.2.5 <i>Filtering by removing single attributes</i>	30
6.3 EXPERIMENT 3	30
6.3.1 <i>Filtering on rule coverage</i>	31
6.3.2 <i>Filtering on rule probability</i>	33
6.3.3 <i>Filtering on reduct length</i>	33
6.3.4 <i>Filtering on reduct support</i>	34
6.3.5 <i>Filtering by removing single attributes</i>	34
6.4 EXPERIMENT 4	34
6.4.1 <i>Filtering on rule coverage</i>	35
6.4.2 <i>Filtering on rule probability</i>	36
6.4.3 <i>Filtering on reduct length</i>	36

6.4.4 Filtering on reduct support	37
6.4.5 Filtering by removing single attributes	37
6.5 EXPERIMENT 5	37
6.5.1 Filtering on rule coverage.....	38
6.5.2 Filtering on rule probability	39
6.5.3 Filtering on reduct length	39
6.5.4 Filtering on reduct support	40
6.5.5 Filtering by removing single attributes	40
6.6 EXPERIMENT 6	40
6.7 EXPERIMENT 7	42
7. DISCUSSION AND CONCLUSION	43
8. BIBLIOGRAPHY	45
APPENDIX	47

Preface

This Diploma thesis was submitted to the Faculty of Physics, Informatics and Mathematics, Department of Computer and Information Science in partial fulfillment of the requirements for the degree «Sivilingeniør» at the Norwegian University of Science and Technology.

Acknowledgments

First of all I would like to thank my fiancée, Synnøve Støre, for supporting and bearing with me through the time of working with this Diploma thesis.

I would like to thank my formal supervisor, Associate Professor Mihhail Matskin, for help with administrative concerns.

Thanks to Ph.D. student Aleksander Øhrn for helping me with ROSETTA and generally with my work. Also, thanks to Ph.D. student Staal Vinterbo and Tor-Kristian Jenssen for interesting discussions.

Last, but not least, I would like to thank Professor Jan Komorowski first of all for giving me the assignment, and also for helping and supporting me during the work.

I would like to thank once more my co-authors in the paper [CKØ98], Jan Komorowski and Aleksander Øhrn for joining in writing my first publication. They were both very helpful in the process.

Trondheim, February 23, 1998

Ulf Carlin

1. Introduction

Acute appendicitis is one of the most common problems in clinical surgery in the western world [Eri96, GFI94]. The diagnosis is difficult even for experienced surgeons, and it has to be balanced between two types of diagnostic errors. On one side is the possibility of performing an unnecessary operation. On the other hand, a delayed diagnosis may lead to perforation of the appendix. Since perforation of the appendix leads to morbidity and occasionally death, a high rate of unnecessary surgical interventions is usually accepted. It is therefore valuable to perform computational analysis of collected data with the objective of improving various aspects of diagnosis.

A database consisting of 257 acute appendicitis patients were collected by Hallan et al. [HÅE97a, HÅE97b]. The data was analysed using logistic regression, and the following two conclusions were drawn:

1. Logistic regression computer models performed approximately equally well as experienced surgeons when using only clinical data in the diagnosis of acute appendicitis.
2. Biochemical tests are of additional value in a logistic regression computer model for diagnosing acute appendicitis.

In this thesis, the database has been reanalyzed using the rough set approach [Paw82, Paw91]. Some results from the rough set analysis is presented in [CKØ98]. The rough set analysis has been done with two important objectives.

First of all, the rough set analysis of the data was performed with the hope of improving the diagnosis of acute appendicitis with the help of a rough set classifier consisting of a set of propositional rules. An additional goal was to extract new knowledge from the database.

Secondly, the rough set approach was to be tested against and compared to a logistic regression analysis on a domain where the logistic regression analysis performed very well. If the rough set approach is to be accepted by other science communities (especially the medical community) it has to perform at least as well as well-known statistical methods (e.g. logistic regression).

The rough set approach takes data in the format of a simple relational table as input. After some preprocessing, reduction of information in the form of so-called reducts is done. From one or a collection of such reducts, simple decision rules are generated. There are possibilities for removing reducts and rules that are supposed not to perform well. The collection of rules obtained through this process may then be used as a classifier. The process of generating rough set classifiers is typically an iterated waterfall cycle with fine-tuning and possible backtracking on the individual substeps. A more detailed description of the steps involved in computing such a classifier comes later in this thesis.

The results of the present rough set analyses were broadly speaking the following. Rough set classifiers in the form of rule bases that performed equally well as the logistic regression models were found. The conclusions by Hallan et al. were supported by the rough set analysis. In addition, single rules that performed especially well were identified. These rules could possibly constitute new knowledge to the field of diagnosing acute appendicitis. In order to be accepted as new knowledge, the rules should be evaluated by medical experts first.

One of the greatest advantages of the rough set approach is the simple explainable rules that makes it possible to identify new knowledge from a database.

One problem that occurred in the rough set analysis was that the best performing models consisted of thousands of rules. A rule base consisting of that many rules has lost the advantage of easily inspectable rules, as it is infeasible to inspect every single rule for possible new knowledge when the rule bases become too large. When rules were filtered away (on different criteria), however, the rule sets became manageable, but not without a loss in the performance.

1.1 Data mining and knowledge discovery

Both logistic regression and rough set analysis can be classified as so-called data mining methods, among many other techniques from the fields of statistics, mathematics, pattern recognition, machine learning and databases. Data mining is a subtask in the field of knowledge discovery in databases (KDD). In [FPSS96] KDD is defined as «the nontrivial process of identifying valid, novel, potential useful, and ultimately understandable patterns in data.». Thus, KDD is the process of discovering useful knowledge from data.

While data mining is concerned with the specific information or pattern extraction from data, is KDD referring to the entire iterative and interactive process of data preparation, data selection, cleaning, preprocessing, information extraction (data mining), and interpretation of the extracted information into knowledge. Here, the term data means non-interpreted atomic pieces of information, a pattern is an expression in some language describing a subset of the data or a model applicable to the subset, while knowledge is some higher understanding about the properties of the data set as a whole, including dependencies between such properties. The knowledge can for instance be expressed as simple propositional rules as in the rough set approach. For a detailed description of the steps of KDD, the reader is referred to [FPSS96].

Data mining tools have a great potential of finding new and useful knowledge hidden in databases of today (for instance medical databases). The data mining techniques are typically superior to humans when there are a great number of parameters and objects to consider and the dependencies between them are complex. In addition, data mining techniques have a greater possibility for objectivity than when the data are studied by humans. There is typically an inevitable bias introduced when data are studied by human experts, because their analyses are typically colored by their former knowledge. Still, most data mining methods also have some kind of built-in bias.

1.2 Intended Reader

Intended reader of this thesis are physicians and scientist in the data mining/rough set community. Also scientist with a general interest in extracting knowledge from structured data may find the methodology applied in this thesis interesting, even though the concrete data studied is from the medical domain.

This thesis can be read without any former knowledge about the specific medical domain under consideration, acute appendicitis. The reader should, however, have some basic knowledge of discrete mathematics. Basic knowledge of standard mathematical notation is thus assumed.

1.3 The Structure of the Thesis

This thesis consists of two main parts. The first part, including Chapter 2 and 3, gives a theoretical foundation for the experiments. The second part, including Chapter 4, 5, 6, 7, and 8, consists of description of the data, experimental method, the actual experiments, discussion of the obtained results, and conclusion.

In Chapter 2 some concepts from the rough set theory is described. The steps in computing a classifier of propositional rules is described, and finally the software system ROSETTA is shortly presented.

Chapter 3 describes techniques for evaluating a model, and some measures that can be used in assessing the performance of the model.

The data material is presented in Chapter 4, together with some simple introductory analyses.

Chapter 5 presents the general experimental methodology used in this thesis.

In Chapter 6, the results of the experiments are presented.

The results obtained are discussed in Chapter 7, and some conclusions are drawn.

2. Preliminary Notions: Rough Sets

This section explains some basic notions from the rough set theory necessary for building a classifier consisting of simple propositional rules. Rough set theory [Paw82, Paw91] was introduced in the early eighties as an alternative mathematical tool to deal with uncertainty in artificial intelligence applications. It has previously been applied to the medical domain (see for instance, [Slo88, Slo92, ØVSK97, VOF98]).

2.1 Information Systems, Decision Systems and Indiscernibility

An *information system* is a tuple: $\mathbf{A} = (U, A)$. Here, U is a non-empty finite set of objects called the *universe*. A is a non-empty finite set of *attributes*. Each attribute a in A is defined by an attribute function: $a : U \rightarrow V_a$, where V_a is the set of values for the attribute a . An information system can be seen as a data table where columns are labeled by attributes, rows are labeled by objects and the entries in the table correspond to attribute values for the different objects. A *decision system* is an information system of the form $\mathbf{A} = (U, A \cup \{d\})$. Here, $d \notin A$ is a special attribute called the *decision attribute*. The attributes in A are called the *conditions*.

In a medical domain, as the one analyzed in this thesis, the universe typically consists of patients (the objects). The attributes can be different medical measurements, clinical variables, laboratory tests, symptoms, and other information that might be relevant for the diagnosis of the disease. The decision attribute is typically some (*a posteriori*) diagnosis.

Let \mathbf{A} be an information system and $B \subseteq A$. Then we can define an equivalence relation between objects in the universe by: $IND(B) = \{ (x,y) \in U \times U \mid \forall a \in B a(x)=a(y) \}$. This relation is called the *B-indiscernibility relation*; it gives a partitioning of the universe in equivalence classes, where objects in each equivalence class are indiscernible from each other, with respect to the attributes in B . The equivalence class to which an object $x \in U$ belongs, is denoted $[x]_{IND(B)}$.

2.2 Reducts

Any minimal $B \subseteq A$ such that $IND(A) = IND(B)$ is called a *reduct* in the information system \mathbf{A} . The set of all reducts in \mathbf{A} is denoted by $RED(\mathbf{A})$. A reduct preserves the partitioning of the universe, and hence the ability to perform classifications. A reduct constitutes a discernibility-preserving elimination of irrelevant information. Reducts are computed with the help of a *discernibility matrix* and a *discernibility function* [SR92].

Let there be n objects in the information system \mathbf{A} , that is $|\mathbf{A}| = n$. Then the *discernibility matrix* of \mathbf{A} , denoted $M(\mathbf{A})$, is an $n \times n$ matrix with elements c_{ij} , where c_{ij} is defined as:

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \quad \text{for } i, j = 1, \dots, n$$

The entry c_{ij} in $M(\mathbf{A})$ consists of all the attributes which discern objects x_i and x_j . Note that $M(\mathbf{A})$ is a symmetric matrix.

From the discernibility matrix we can compute the *discernibility function*. It is a Boolean function of m Boolean variables a_1^*, \dots, a_m^* corresponding to the attributes a_1, \dots, a_m , respectively:

$$f_{\mathbf{A}} : Bool^m \rightarrow Bool$$

The discernibility function is defined as follows:

$$f_{\mathbf{A}}(a_1^*, \dots, a_m^*) = \bigwedge \{ \bigvee c_{ij}^* \mid 1 \leq i < j \leq n, c_{ij} \neq \emptyset \} \quad \text{where } c_{ij}^* = \{a^* \mid a \in c_{ij}\}$$

It can be shown [SR92] that the set $RED(\mathbf{A})$ can be determined by the set of all *prime implicants* of $f_{\mathbf{A}}$, that is:

$$\begin{aligned} a_{i_1}^* \wedge \dots \wedge a_{i_k}^* & \text{ is a prime implicant of } f_{\mathbf{A}} \\ \text{iff} \\ \{ a_{i_1}, \dots, a_{i_k} \} & \in RED(\mathbf{A}) \end{aligned}$$

A prime implicant of a Boolean expression is an expression that is simpler than the original, but the truth of which implies the truth of the original expression. The set of prime implicants of a logical expression is the set of disjuncts in the function written in disjunctive normal form.

Computing reducts is a very time consuming task when there are either too many objects, attributes or attribute values. The computational complexity of computing a minimal reduct has in fact shown to be NP-hard [SR92]. In cases where the decision table is too large to compute exact reducts, approximate algorithms for reduct computation can be used. These approximate algorithms can of course not generally find all reducts and give optimal solutions. Examples of such approximate algorithms for reduct computation are the genetic algorithm described in [Wro95] and the Johnson algorithm. Both are implemented in ROSETTA (See Section 2.6).

The Johnson algorithm finds a single reduct by a simple greedy algorithm. First it finds the attribute that occurs in the most alterms in the discernibility function. Then this attribute is added to the reduct, and the alterms containing the attribute is deleted. These steps are repeated until there are no alterms left.

2.3 Other Reduct Types

There exist a number of different types of reducts, and basically they differ only in how the discernibility function is constructed from the discernibility matrix. The reduct type defined above is sometimes called a full reduct. It is strictly defined, and single «atypical» objects may prevent finding general patterns.

2.3.1 Object-related Reducts

When computing so-called *object-related reducts*, the focus is turned to discerning one object from the other objects. For each object, the minimal sets of attributes that discern that object from all other objects are computed. The resulting set of reducts constitutes the object-related reducts for the decision table. Rules generated from object-related reducts are typically shorter than rules generated from ordinary reducts. One hopes that this leads to more "noise tolerant" and general rules.

To compute object-related reducts we are only interested in those attributes which discern one object from the rest. This information lies in the corresponding column (or row) in $M(\mathbf{A})$. Thus f_{Ax_k} (the discernibility function for object k, x_k) can be constructed as:

$$f_{Ax_k}(a_1^*, \dots, a_m^*) = \wedge \{ \vee c_{ik}^* \mid 1 \leq i \leq n, c_{ik} \neq \emptyset \} \quad \text{where} \quad c_{ik}^* = \{ a^* \mid a \in c_{ik} \}$$

The connection to the (ordinary) discernibility function is as follows:

$$f_{\mathbf{A}}(a_1^*, \dots, a_m^*) = \wedge \{ f_{Ax_k}(a_1^*, \dots, a_m^*) \mid 1 \leq k \leq n \}$$

Note that using this formula all alterms in the first definition of the discernibility function appear twice, as *all* entries in the *symmetric* discernibility matrix $M(\mathbf{A})$ are included in the above formula. The formula is therefore usually not used in practice. The duplicates can of course be removed in the process of computing prime implicants using the property of idempotence for conjunction.

2.3.2 Dynamic Reducts

Another method to achieve more "noise tolerant" reducts is *dynamic reduct* [BSS94] computation. Dynamic reducts are computed by finding reducts for sampled subtables of the original decision table $\mathbf{A} = (U, A \cup \{d\})$. Any decision table of the form $\mathbf{B} = (U', A \cup \{d\})$ such that $U' \subseteq U$ is called a subtable of \mathbf{A} . The reduct computation process is repeated for a certain number of different samples, often of different sizes. The most «stable» reducts (the most frequent) from the sampled subtables are returned and these reducts are called dynamic reducts. In the process of computing dynamic reducts, one may use any reduct computation strategy, for instance full reducts or object-related reducts.

The random sampling of subtables is done with the hope that «atypical» or «noisy» objects at least sometimes are left out in the sampling process. This may then lead to that underlying general patterns are found, that otherwise would have been obscured by the outliers.

The type of dynamic reducts considered here are the so-called (F, ϵ) -*generalized dynamic reducts* [Baz98]. These dynamic reducts are typically not ordinary reducts of the complete table, but actual reducts of some sampled subtables. The rules generated from these reducts are likely to be inconsistent, but are believed to be more tolerant to noise and capture more general patterns in the data than ordinary reducts.

Sampling from small subtables produces potentially more approximate and less correct rules (wrt. the training set). Even though, these rules possibly generalize better than rules generated from ordinary reducts.

One method for computing dynamic reducts is described in the following. A random set of subtables is sampled from the decision table. An example of a sampling strategy is to generate 10 samples on each of 5 sampling levels with sizes from 50% to 90% of the original decision table. With this, the following samples of the decision table are meant:

- 10 samples with size 50% of the decision table,
- 10 samples with size 60% of the decision table,
- 10 samples with size 70% of the decision table,
- 10 samples with size 80% of the decision table,
- 10 samples with size 90% of the decision table.

Reducts are then calculated for each of the 50 new decision tables generated in the sampling step.

The reducts may then be returned after a filtering step. This step involves removing reducts that are supposed to be «weaker» than the other reducts. When computing (F, ϵ) -generalized dynamic reducts as described in [Baz98], the filtering step involves removing reducts with a *stability coefficient* lower than some threshold. The stability coefficient for a reduct R is equal to the number of subtables where R is a reduct divided by the total number of subtables in the dynamic reduct process.

One might think of other filtering strategies than filtering on the reduct stability coefficient. Reducts can also be filtered according to their performance [VOF98]. A separate part of the testing decision table can be held back in the (dynamic) reduct computation process. Rules may then be generated from the reducts and the table they were computed from. The reducts can then be ranked according to how well their corresponding rule set performs on the part of the table held back. There are different possibilities for ranking the reducts. They can for instance be ranked according to accuracy (or equivalently error rate), sensitivity, specificity or any combination of such performance measures (See Section 3.3.1). The reducts that are ranked below some threshold may then be removed.

The *rules* may also be filtered. This can be done for example on rule length, rule support, rule coverage or rule stability coefficients (See Section 2.5). Reduct performance filtering and the other types of filtering strategies mentioned here are implemented in the ROSETTA software system (See Section 2.6).

2.4 Reducts Modulo the Decision Attribute

All reducts described to this point are reducts in the context of a general information systems. Reducts can also be defined in the context of decision tables, taking the decision attribute into account.

When reducts are to be used for classification, one is typically not interested in discerning between objects belonging to the same decision class. The only thing that has to be done to achieve this is to delete the alternants from the discernibility function that discerns between objects with the same decision class. The resulting reducts are minimal sets of attributes that enable one to make the same classification as the whole set of attributes. They are often called reducts relative to the decision attribute or reducts modulo the decision attribute.

For inconsistent decision systems, some additional notions as e.g. generalized decision must also be introduced, [Pawlak 91].

2.5 Decision Rules

In this section the way ROSETTA computes decision rules from reducts is described. For a formal description of decision rules, see [Sko93]. A set of decision rules can collectively form a classifier that can classify new objects.

An expression of the form $a=v$ such that $a \in A \cup \{d\}$ and $v \in V_a$ is called a descriptor. A decision rule (also called classification rule) is an expression built from descriptors in the following way:

$$(a_{i1} = v_{i1}) \wedge (a_{i2} = v_{i2}) \wedge \dots \wedge (a_{ik} = v_{ik}) \Rightarrow d = v$$

Here, $a_{i1}, \dots, a_{ik} \in A$, v_{i1}, \dots, v_{ik} are values for conditional attributes, d is a decision attribute and v is a value for the decision attribute.

When reducts have been computed, it is trivial to generate the decision rules. The computational effort lies in the reduct computation. Decision rules are generated simply by laying each reduct over the originating decision table, and reading off the attribute values as given in the following pseudo-code:

```
For each reduct  $R = \{a_1, \dots, a_p\}$  ( $p \leq \text{card}(A)$ ) do
  For each object  $x \in U$  do
    construct the decision rule  $(a_1=a_1(x)) \wedge \dots \wedge (a_p=a_p(x)) \Rightarrow d = d(x)$ 
```

Here $a_i(x)$ is the value of attribute a_i for object x . The rules describe the objects in the decision table from which the reducts were computed. One hopes that these rules also will describe future objects reasonably well. Note that inconsistent rule sets may be generated. Two rules may for instance have the same antecedent but different consequences.

When the rules are generated, the number of objects that generate the same rule is typically recorded. This number is called the rule's *support count*. The support count can be useful when the rules are going to be used to classify new objects. From the support count some quantitative measures as the *coverage* and the *stability coefficient* can be computed. The coverage denotes the fraction of objects in the decision table that the rule covers, that is the fraction of objects with the same decision as the rule that «matches» it. The stability coefficient [Baz98] is a measure of how «stable» a rule generated by means of dynamic reduct computation is.

When classifying a new object using a set of rules, the rule base is searched for rules that match the object's attributes (rules that «fire»). If all matching rules suggest the same decision, that decision is chosen for the new object. If no rules match, a fallback decision may be chosen. A typical fallback choice is the most frequently occurring decision class.

If rules with different decisions match the new object, some kind of voting must take place to resolve the conflict. An election procedure can be simulated among the matching rules, where each rule gets a number of votes according to its support. Typically the decision class with the largest number of votes would be selected, but it is

possible to prioritize a decision class by letting all objects with a voting percentage for class C above some threshold be classified as such. By varying this prioritization threshold, points on a ROC curve (as described in Section 3.3.2) can be generated. The area under the ROC curve may then be computed using for instance the trapezoidal integration rule.

2.6 The ROSETTA system

The tool for doing rough set data mining in this thesis is the ROSETTA software system [ØK97, ØKSS98].

ROSETTA is a system for data mining based on the rough set theory. ROSETTA is a short form of Rough Set Toolkit for Analysis of Data. It consists of a computational kernel and a GUI (Graphical User Interface) front-end. From the front-end all the functionality of the kernel is easily available. ROSETTA runs on PCs operating under Windows NT or Windows 95.

ROSETTA has been developed as a co-operation between the Knowledge Systems Group at the Norwegian University of Science and Technology and the Logic Group at Warsaw University. The kernel architecture, GUI front-end and computational kernel is designed and implemented at the Knowledge Systems Group, while sections of the computational kernel is based on parts of RSES (Rough Set Expert System) from the Logic Group.

ROSETTA supports every step in generating a classifier consisting of decision rules from a simple relational table as described earlier in this chapter. This includes (among other things): table completion, attribute discretization, reduct computation (full, object-related, dynamic, genetic, Johnson algorithm), reduct filtering, rule generation, rule filtering, and classification of new objects. In addition ROSETTA supports n-fold cross-validation and the possibility of automation of lengthy and repetitive sequences of tasks in a command script feature.

A limited version of ROSETTA is publicly available for non-commercial use at the URL:
<http://www.idi.ntnu.no/~aleks/rosetta/rosetta.html>

3. Model Evaluation

When a model (e.g. classifier) has been built by any data mining technique, it needs to be evaluated. One wants to estimate how well the model performs on future examples on average. The process of validation consists of calculating some performance measure that reflects the model's ability to classify objects correctly (as e.g. error rate or accuracy). One typically wants to test a model on different data than it was derived from. When only one data set is available, which usually is the case, a strategy for dividing the original set of objects has to be decided on.

3.1 Simple Validation

Simple validation (often called the holdout method) is carried out by setting aside a part of the data set for testing. The rest of the data set, usually called the training set, is used to build the model. The division is done at random and the relative size between the training and test sets may vary. None of the sets should be too small, however. Testing on the training data should be avoided, as this gives a poor estimate of the generalization ability of the method. It promises much better performance than will be obtained with independent test data.

3.2 Cross-validation

In *cross-validation* the data set is randomly divided into two equally large sets. On each half, a model is built, and subsequently tested on the other half. The average performance measure is returned as an estimate of the true performance of a model generated from all of the data.

A generalization of cross-validation is *k-fold cross-validation* (CV). The intention of k-fold CV is to avoid or at least reduce the possible bias introduced by relying on any one particular division into test and training sets. In this method the data set is randomly split into k mutually exclusive subsets of approximately equal size (the folds). k iterations are then executed on the data. In each iteration k-1 subsets are used as training set and the last subset is used as test set in such a manner that each subset is guaranteed to be in the training set k-1 times and in the test set once. The average computed performance measure from the k iterations is returned as an estimate of the performance of a model built from all of the data. With increasing value of k, one gets training sets that are almost as big as all of the data, and thus hopefully good estimates of the performance. If k equals the number of objects in the data table, we get *leave-one-out* (LOO) CV.

The advantage of CV is that all of the data is used for both training and testing. A disadvantage is the amount of computing required. k-fold CV requires approximately k times as much computing as simple validation. Because of this k-fold CV with a high value for k may be infeasible to compute in practice. On the other hand, if k gets too small, the performance measure is pessimistically biased because of the small training sets in the CV analyses. Because of this, a moderate value for k usually is preferred. Common choices are k=5 or k=10.

Cross-validation is supported by ROSETTA in the command script feature, see ROSETTA's on-line help.

3.3 Test Performance Statistics

In this section different strategies for measuring the performance of a model on a test set is discussed. The discussion will be done with the medical domain as example. Still, the presented methods may be used for any domain. One important requirement is that the concept to be learned by the classifier is binary, i.e. has two values.

3.3.1 Simple Measures from the Contingency Matrix

5 performance measures that describe the ability of a test to diagnose the true patient status will be presented in this section. These measures of performance are automatically given when classifying new sets of objects in ROSETTA when both the real diagnosis and the test have two outcomes.

The following two outcomes are typically possible for the diagnosis: Either the patient has a disease, which will be called a positive disease status, or the patient has not the disease, which will be called a negative disease status. Similarly, a test that predicts the disease in the patient will be called a positive test, and conversely a test predicting absence of the disease will be called a negative test. A test might for instance be the presence or absence of some symptom, a computer model prediction or a biochemical test.

As an illustration of the accuracy of a test, the 2×2 contingency matrix in Table 1 can be set up.

		Test		
		Negative	Positive	
Disease status	Negative	True negatives (<i>a</i>)	False positives (<i>b</i>)	Specificity
	Positive	False negatives (<i>c</i>)	True positives (<i>d</i>)	Sensitivity
		Negative predictive value	Positive predictive value	Accuracy

Table 1: Contingency matrix

a, *b*, *c*, and *d* are frequencies for the events in the corresponding entries. From the matrix the 5 performance measures shown in the last row and column can easily be computed.

Two measures have to do with the proportions of the persons with positive and negative disease status, respectively, that are correctly diagnosed by the test.

Sensitivity is the proportion of the patients with positive disease status who are correctly identified by the test. *Specificity* is the proportion of the patients with negative disease status who are correctly identified by the test.

These two measures are frequently used in describing a diagnostic test seen from the direction of the actual disease status. But in clinical practice the test result is all that is known, and the whole point of a diagnostic test is to use it to make a diagnosis. Therefore it might be interesting to know what the probability of the test giving the correct diagnosis is, whether the test is positive or negative. This is done by approaching the data from the direction of the test results.

Positive predictive value is the proportion of patients with negative test result who are correctly diagnosed. *Negative predictive value* is the proportion of patients with positive test result who are correctly diagnosed.

Similarly the overall *accuracy* is of course the proportion of patients at all who are correctly diagnosed by the test.

The five performance measures just presented can be computed in the following way. The letters refer to the frequencies in Table 1.

$$Sensitivity = d/(c+d)$$

$$Specificity = a/(a+b)$$

$$Positive\ predictive\ value = d/(b+d)$$

$$Negative\ predictive\ value = a/(a+c)$$

$$Accuracy = (a+d)/(a+b+c+d)$$

The complement of the accuracy is the *error rate*, which is a frequently used test performance statistic in the data mining literature:

$$Error\ rate = (b+c)/(a+b+c+d) = 1 - Accuracy$$

Sensitivity and specificity are sometimes called true-positive and true-negative rates, respectively. These names are ambiguous as they easily can be confused with the predictive values.

The disadvantage with the sensitivity and specificity is that they do not assess the accuracy of the test in a clinically useful way. An advantage with the sensitivity and specificity is that they are not affected by the proportion of subjects with the disease, which is called the prevalence. The negative and positive predictive values, on the other hand, are clinically useful, but they depend strongly on the prevalence in the data material. Because of this, the sensitivity and specificity can be used to compare results of tests on patient groups with different prevalences. The predictive values observed in a sample should on the other hand not be taken as applying universally, because the prevalence of disease might be totally different.

A typical situation is when the prevalence in the sample group is higher than in the rest of the population. A lower prevalence results in that positive disease status is more uncommon. This leads to that we can be more sure that a negative test indicates no abnormality, and similarly less sure that a positive test result really indicates an abnormal patient.

So far only binary outcomes from a test have been considered. A common situation arises when the (two valued) diagnosis is to be made using a test that takes on more than two values or even a continuous range of values. When faced with such continuous test outputs, it is necessary to define a cut-off value in order to count the number of true and false predictions to build the 2×2 contingency table. Since this definition is essentially arbitrary, the best procedure is to study the effect of different cut-off values upon the performance statistics. It is then possible to find the combination that gives the highest accuracy or the best combination of sensitivity and specificity, for example by maximizing the sum. Still, to choose the best cut-off value is not a trivial decision. The relative costs (not necessarily financial) associated with false positive and false negative test results should be taken into consideration. Therefore it might not always be optimal for example to look for the best accuracy or maximizing the sum of sensitivity and specificity. However, if the "cost" of a false positive result is the same as the "cost" of a false negative test, the best cut-off value is that which maximizes the sum of the sensitivity and specificity.

3.3.2 ROC analysis

A graphical approach to finding the best cut-off value is to plot $1 - \textit{specificity}$ (false-positive rate) on the x-axis versus the *sensitivity* (true-positive rate) on the y-axis for each possible cut-off value (decision threshold). The curve obtained by joining the points is known as a "receiver operating characteristic" curve, or ROC curve for short [HM82, ZC93]. The point on the ROC curve nearest the point (0,1) can now be used as the "best" combination of sensitivity and specificity.

A ROC curve is however not primarily used to find the «best» cut-off value. It is actually a very good visual representation of the accuracy of a test. It shows how well a model represents observed data independent of a particular choice of a cut-off value. It is very useful in comparing the performance of different models, as for instance logistic regression models and rough set models. Any model giving a continuous or multivalued prediction score when classifying new objects is suitable to be subjected to a ROC analysis.

A typical ROC curve is shown in Figure 1. Since a test or model should have high sensitivity and specificity for all cut-off values, a curve that is well above the diagonal line is a good test. The diagonal line represent a test without the ability to discriminate. Along the diagonal the true- and false-positive rates are equal. A system can achieve such a performance by chance alone.

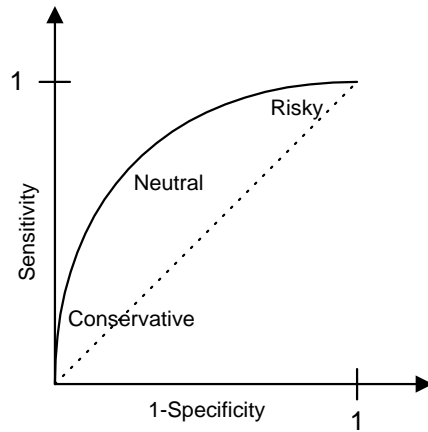


Figure 1: Typical ROC curve

Points on the ROC curve close to the point (0,0) correspond to conservative decision thresholds. Thresholds resulting in such points have high specificity but low sensitivity. A threshold giving such a point is conservative in the sense that the test (with the particular threshold) is cautious in classifying an object as positive («sick»). Analogously, points close to (1,1) correspond to risky decision thresholds. These thresholds have high sensitivity, but low specificity, which means that the corresponding test is generous in classifying objects as positive, resulting in a high rate of false positives. In most applications (where the "cost" of a false positive result is approximately the same as the "cost" of a false negative test) tests (with corresponding decision threshold) in the neutral area in Figure 1 are preferred.

Comparing curves can sometimes be laborious and difficult. Therefore a single number could be useful to compare models. There exist many different measures associated with ROC curves. One popular measure usually employed is the area under the ROC curve (AUC). It summarizes a curve in one single number. Of course there is necessarily a loss of information in the process of computing the AUC. Still the AUC reflects fairly well the «goodness» of a ROC curve.

The AUC will vary between 0.5 and 1. If an AUC less than 0.5 is computed (i.e. the curve is below the diagonal), it is always possible to reverse the decision strategy to get an AUC higher than 0.5. The area under the ROC curve should be as close to 1 as possible. An area under the ROC curve of 0.5 denotes no discrimination at all, and a value approaching 1 signifies excellent discrimination.

In Figure 2 the ROC curves of two imaginary tests are shown. Test A has higher levels of sensitivity at all levels of specificity. The superiority of test A compared with test B is adequately expressed by the difference in the areas under the two curves.

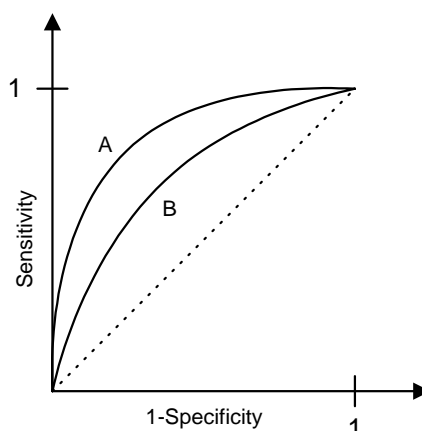


Figure 2: Two ROC curves with different AUC

Situations where the AUC does not give enough information exist, however. In Figure 3 it is hard to decide which curve is the best. The AUC for the two tests are equal. Again the relative cost of false positive and false negative test results is vital. This time it is in the decision of which test is best. If false positives are to be avoided, the specificity should be higher and test B would be the preferred one. Conversely, test A would be chosen if false negatives were to be avoided.

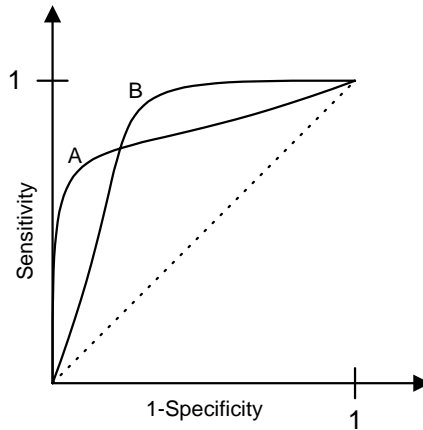


Figure 3: Two ROC curves with the same AUC

There are two important properties that make ROC analysis a very good measure of a test system's accuracy. Firstly, the ROC curve is built from pairs of sensitivity and specificity, and the analysis is therefore independent of the prevalence as discussed in the preceding section. Secondly, it is independent of the decision bias, that is a model's particular tendency to choose one decision over another; A ROC analysis is independent of any particular choice for the cut-off value between the two possible diagnoses, as *all* cut-off values are taken into account in the analysis.

The AUC is closely related to some well-known statistics. The AUC computed with the trapezoidal rule is in fact identical to the well-studied nonparametric *Wilcoxon statistic* [HM82]. There exist also an intuitive meaning of these two indices: They are both identical to the probability of a correct ranking of a (healthy, sick) pair of patients. With this the following is meant: The two indices measure the probability that in a randomly pair of patients, one having a positive diagnosis and the other having a negative diagnosis, the test prediction of the two patients will allow them to be correctly identified [HM82]. To exemplify this, an AUC area of 0.8 for a test means that a randomly chosen individual from the diseased group has a test value larger than that for a randomly chosen individual from the healthy group with probability 0.8.

The AUC is also related to the *overlap index* of Hartz by the formula: $area = 1 - (overlap\ index/2)$ [ZC93].

4. Data Material and Introductory Analyses

4.1 Data Material

The data material studied in this thesis consists of 257 patients with suspected acute appendicitis. Originally, 309 patients were recorded between March 1992 and January 1994, but 52 patients were excluded from the analysis because of missing values on one or more of the variables. The data set has previously been analyzed using logistic regression [HÅE97a, HÅE97b].

For each patient the attributes listed in Table 2 and Table 3 were recorded. The binary and numerical attributes are summarized in Table 2 and Table 3, respectively.

Attribute	Description	Statistics		Correlation w/ DIAGNOSIS
		yes % (count)	no % (count)	
SEX	Male sex?	55.3 (142)	44.7 (115)	0.288
ANOREXIA	Anorexia?	69.3 (178)	30.7 (79)	0.176
NAUSEA	Nausea or vomiting?	70.8 (182)	29.2 (75)	0.099
PREVIOUS MOVEMENT	Previous surgery? Aggravation of pain by movement?	9.3 (24)	90.7 (233)	-0.059
COUGHING	Aggravation of pain by coughing?	61.5 (158)	38.5 (99)	0.210
MICTUR	Normal micturition?	59.9 (154)	40.1 (103)	0.250
TENDRLQ	Tenderness in right lower quadrant?	87.2 (224)	12.8 (33)	-0.034
REBTEND	Rebound tenderness in right lower quadrant?	86.0 (221)	14.0 (36)	0.317
GUARD	Guarding or rigidity?	55.3 (142)	44.7 (115)	0.384
CLASSIC	Classic migration of pain?	30.7 (79)	69.3 (178)	0.276
LEFT	Shift to the left in differential count?	49.4 (127)	50.6 (130)	0.410
DIAGNOSIS	(Final diagnosis:) acute appendicitis?	53.3 (137)	46.7 (120)	0.398
		38.1 (98)	61.9 (159)	1.0

Table 2: Binary attributes

Attribute	Description	Unit	Statistics			Correlation w/ DIAGNOSIS
			Mean (SD)	Median	Range	
AGE	Age	years	26.8 (17.0)	22	3-86	0.126
DURATION	Duration of pain	hours	35.3 (53.8)	22	2-600	-0.070
TEMP	Rectal temperature	°C	37.8 (0.75)	37.7	36.4-40.3	0.179
ESR	Erythrocyte sedimentation rate	mm	14.1 (15.8)	10	1-90	0.131
CRP	C-reactive protein concentration	mg/L	32.8 (48.7)	12	0-260	0.297
WBC	White blood cell count	$\times 10^9$	12.3 (4.79)	12.1	2.9-31.0	0.458
NEUTRO	Neutrophil count	%	77.1 (11.4)	80	38-93	0.377

Table 3: Numerical attributes

All patients with acute abdominal pain referred consecutively to the department of surgery at Innherred Hospital by general practitioners were part of a study. Innherred Hospital is a district general hospital in Norway serving 90 000 inhabitants. An initial clinical examination was performed on each patient in the emergency department. If acute appendicitis was one of the possible different diagnoses, the surgeon performing the examination filled in a data collection sheet containing the clinical attributes in Table 2 and Table 3. Delayed diagnoses of acute appendicitis were not included in the data material.

At the initial examination blood was drawn for analysis of inflammatory parameters. These parameters were: ESR, CRP, WBC, NEUTRO, and LEFT. The LEFT attribute was not a part of either of the articles by Hallan et al. [HÅE97a, HÅE97b]. At the initial examination (and thus before the results of the blood test) the surgeon collecting the data estimated a probability of appendicitis for the patient. See Section 6 for an analysis of these probability estimations.

In addition to the attributes shown in Table 2 and Table 3 and the probability estimates, two variables were collected. These variables were not a part of the experiments by Hallan et al., and no description of the semantics was acquired when Hallan was contacted. Because of this, the two variables were excluded from all experiments.

The DIAGNOSIS attribute is the (*a posteriori*) decision attribute d in the analysis. It shows which patients actually turned out to have appendicitis. As can be seen in Table 2, 98 patients (38%) turned out to have appendicitis and 159 (62%) turned out to have some other disease or non-specific abdominal pain. The final diagnosis of acute appendicitis was based on histological examination of the excised appendix. Other diagnoses were based on routine investigation with repeated clinical examination, biochemical tests, imaging techniques and, if necessary, surgery.

In the analysis, different subsets of all the attributes will be used as A in the decision system $\mathbf{A} = (U, A \cup \{d\})$. This is done in order to make a fair comparison of the diagnostic ability of the logistic regression model, the rough set model, and the surgeons probability estimate. For further details about the collection of the data material, see [HÅE97a, HÅE97b].

4.1.1 Discretization

The rough set theory is based on the concept of indiscernibility. Numerical attributes should be discretized into intervals, so that numbers falling within the same interval are deemed as being indiscernible. For the majority of the experiments in this thesis, the discretization of the numerical attributes in Table 3 shown in Table 4 was chosen.

Attribute	Intervals	Count	Description
AGE	$[-\infty, 17)$	84	Low
	$[17, 31)$	90	Middle
	$[31, \infty)$	83	High
DURATION	$[-\infty, 13)$	96	Short
	$[13, 30)$	71	Middle
	$[30, \infty)$	90	Long
TEMP	$[-\infty, 37.5)$	87	Low
	$[37.5, 38.1)$	89	Middle
	$[38.1, \infty)$	81	High
ESR	$[-\infty, 10)$	126	Normal
	$[10, 25)$	99	Slightly raised
	$[25, \infty)$	32	Considerably raised
CRP	$[-\infty, 6)$	103	Normal
	$[6, 40)$	95	Slightly raised
	$[40, \infty)$	59	Considerably raised
WBC	$[-\infty, 10.0)$	91	Normal
	$[10.0, 14.0)$	80	Slightly raised
	$[14.0, \infty)$	86	Considerably raised
NEUTRO	$[-\infty, 75)$	87	Normal
	$[75, 85)$	87	Slightly raised
	$[85, \infty)$	83	Considerably raised

Table 4: Discretization of numerical attributes

The CRP, WBC, and NEUTRO attributes were discretized by a medical expert (S. Hallan). The ESR attribute was discretized with the same intervals as in [OZT93]. The AGE, DURATION, and TEMP attributes were

discretized into 3 intervals, each with approximately the same size (33%). The discretization of the three attributes AGE, DURATION, and TEMP is questionable, as the discretization has not been done without looking at the test set. It would have been more serious to look at the decision when the whole table was discretized.

Also, there exist algorithms for automatic discretization. Such automatic discretization algorithms are implemented in Rosetta. Usual problems with automatic discretization algorithms is that the number of intervals gets very high or that some intervals are much larger than other intervals. High number of intervals typically leads to too specific rules. For instance a key in a relational database is a reduct that perfectly partition the data, still not very useful.

It is important to remember that the discretization has a great impact on the result from a rough set analysis. In this thesis only one discretization of the data was examined.

Discretizing attributes may imply a loss of information. On the other hand is discretization of continuous attributes necessary for rough set analysis. It is important to have in mind that the discretization has great influence on the performance of a classifier built from the data. In this thesis only one discretization of the data was examined.

4.2 Introductory Analyses

In this section some introductory analyses of the data material is done. This is done without using the rough set approach. First the performance of each binary variable is studied. Then the performance of the surgeons' initial diagnoses is described.

4.2.1 Analysis of Each Binary Variable Used on Its Own

In this first analysis the focus is on the performance of each attribute on its own.

In Table 5, the performance of each binary attribute used on its own is presented. They are sorted according to diagnostic accuracy. In addition to the binary attributes the performance of the «baseline» classification strategy is also presented in the table. The «baseline» strategy is to classify all patients as the majority decision class, which in the appendicitis data table is «not appendicitis».

Attribute	Sensitivity	Specificity	Positive predictive value	Negative predictive value	Accuracy
CLASSIC	75.51	66.67	58.27	81.54	70.04
LEFT	78.57	62.26	56.20	82.50	68.48
REBTEND	79.59	59.75	54.93	82.61	67.32
GUARD	46.94	79.25	58.23	70.79	66.93
SEX	73.47	55.97	50.70	77.39	62.65
Baseline	0.00	100.0	Undefined	61.87	61.87
MICTUR	14.28	88.05	42.42	62.50	59.92
COUGHING	75.51	49.69	48.05	76.70	59.53
PREVIOUS	7.14	89.31	29.17	60.94	57.98
MOVEMENT	74.49	46.54	46.20	74.75	57.20
ANOREXIA	79.59	37.11	43.82	74.68	53.31
TENDRLQ	100.0	22.64	100.0	44.34	52.14
NAUSEA	76.53	32.70	41.21	69.33	49.42

Table 5: Binary attributes' performance ranked according to accuracy (values given in %)

The classification is based on that the presence of some attribute value leads to the classification of the patient as having the diagnosis and absence leads to the classification of the patient as not having the diagnosis.

This kind of comparison is undoubtedly unfair. Even though, it gives an indication of the goodness of each attribute. The attributes may have any combination of values for sensitivity and specificity, and a special combination of certain values may turn out to be valuable, even though the accuracy is low.

As an example we can see from Table 5 that TENDRLQ has a sensitivity of 100%, but the low specificity leads to an accuracy that is lower than most of the other attributes. The attribute may be regarded as important, however, because a surgeon finding that the patient does not have tenderness in right lower quadrant (TENDRLQ) may diagnose the patient as not having appendicitis with high certainty (The negative predictive value is 100% when sensitivity is 100%).

A question that comes to mind is then how representative the objects in the data table is for a parent population or universe. The objects are typically considered as being a representative or random sample of some parent population (even though they typically are not). Such statistical consideration are important when drawing general conclusions from a (small) sample represented in a database.

The appendicitis data table has a high prevalence of having acute appendicitis. Prevalence is the proportion of subjects with an abnormality under consideration. This is because the objects in the data table are patients with suspected acute appendicitis.

Another thing is that the ranking in Table 5 does not at all take into account an attribute's performance in combination with other attributes. This is what we hope to take advantage of in the rough set computer models.

It is also important to note that the semantics of the attributes may be different for different surgeons, and thus lead to differences between surgeons, hospitals and countries.

The computer models in the next chapters should at least perform better than the best attribute on its own, however. Therefore the performance of the CLASSIC attribute will be compared to the rough set computer models in the subsequent chapters. The computer models should also perform better than the «baseline» classification strategy.

4.2.2 Analysis of the Surgeons' Classification

For each patient the surgeon estimated the patient's risk of having acute appendicitis in increments of 10% from 0 to 100%. There were nine different surgeons with two to six years of surgical training who participated in this probability estimation. The estimation was done at the time of the initial examination of the patients, and thus before the result of the blood test was ready. The attributes based on the blood test, and thus not available to the surgeon when he performed the probability estimation, are: ESR, CRP, WBC, and NEUTRO. The probability estimates were, of course, neither a part of the logistic regression analyses in [HÅE97a, HÅE97b] nor of the present rough set analysis.

In Table 6 the surgeons' probability estimates are compared to the actual disease status of the patients (attribute DIAGNOSIS).

Surgeons' percentage estimates	Number of patients	
	Appendicitis	Not appendicitis
0	1	9
10	2	41
20	2	26
30	2	16
40	6	16
50	19	15
60	8	8
70	21	15
80	18	7
90	17	6
100	2	0

Table 6: Surgeons' probability estimates of acute appendicitis compared to the actual disease status of the patients

In Table 7 sensitivity, specificity and accuracy of the probability estimates made by the surgeons for different cut-off values has been computed. The patients with an estimated probability value above the cut-off value are classified as having a positive diagnosis and the other patients as having a negative diagnosis.

Cut-off	Sensitivity	Specificity	Positive predictive value	Negative predictive value	Accuracy
0	0.990	0.057	0.393	0.900	0.412
1	0.969	0.314	0.466	0.943	0.564
2	0.949	0.478	0.528	0.938	0.658
3	0.929	0.579	0.576	0.929	0.712
4	0.867	0.679	0.625	0.893	0.751
5	0.673	0.774	0.647	0.794	0.735
6	0.592	0.824	0.674	0.766	0.735
7	0.378	0.918	0.740	0.705	0.712
8	0.194	0.962	0.760	0.659	0.669
9	0.020	1.000	1.000	0.624	0.626

Table 7: Simple performance measures for the surgeons' classification

We can see from the table that the cut-off value that gives the best accuracy is 4. This cut-off value gives an accuracy of 0.75, a specificity of 0.68, and a sensitivity of 0.87. These results will in the sequel be used to compare the surgeons' classification with the rough set computer models' classification.

To discretize a multivalued variable as the estimated probability to a corresponding binary variable like in the above table, does of course imply a loss of information. The information lost lies in the grading of the probability in the ten different possible values. Therefore it is a little unfair to compare the two-valued "variable" with two-valued predictions resulting from the computer models.

On the basis of the 10 pairs of sensitivity and specificity, the ROC curve in Figure 4 has been drawn. The area under the ROC curve is 0.817 when using the composite trapezoid rule. This result will in the sequel be compared with the results of the rough set analyses.

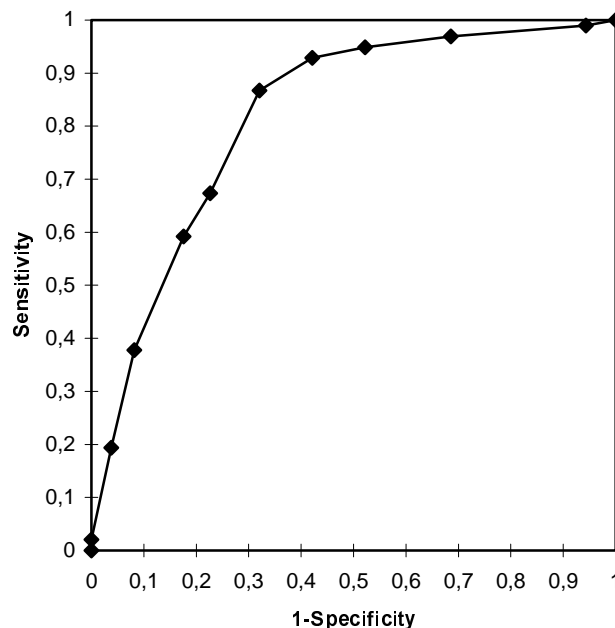


Figure 4: ROC curve reflecting the surgeons' classification

Note that the ROC curve in Figure 4 is based on 10 pairs of sensitivity and specificity. The ROC curves from the computer models in the subsequent section will typically be generated from many more pairs of sensitivity and specificity. Because of the typical convex overall shape of the ROC curves, a curve based on few points joined with straight line segments has typically a smaller area than a curve based on many points.

5. General Methodology

A typical way to carry out KDD experiments is shown in Figure 5. A group of objects to be studied is split in two sets: the training set and the test set. A discovery algorithm is applied on the first set and a classifier is returned. The classifier is then tested on the other set of objects and some kind of performance measure is computed. The process is typically iterated by, for instance, varying the splitting in a systematic fashion (using e.g. cross-validation), and/or adjusting various parameters in the discovery process.

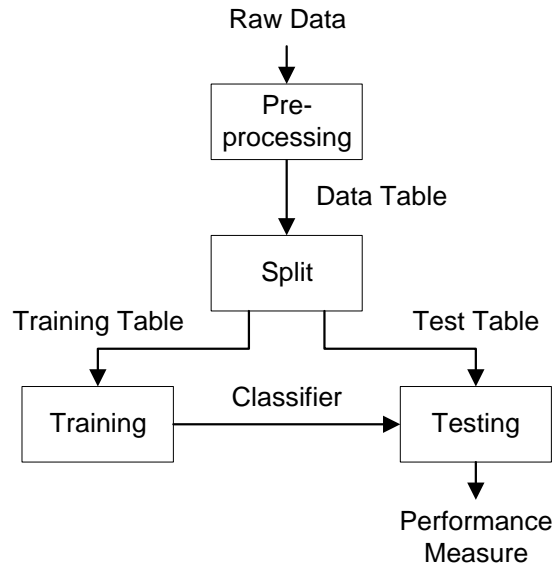


Figure 5: Training/testing cycle

If the rough set approach is used in this general scheme, the diagram in Figure 5 can be decomposed as shown in Figure 6. The rough set approach is typically able to make use of the data stored in a relational database directly. Still, some preprocessing of the data usually should be done. In a rough set analysis the preprocessing box of Figure 5 typically consists of two steps.

The first step is to deal with missing values. To be able to start the analysis, the decision table should be complete. Many strategies for making a decision table complete exist in the ROSETTA software system. In the present analysis the completion algorithm was simply to remove all objects containing missing values. This strategy was chosen by Hallan et al. in their analysis, and since I only had access to the 257 complete objects from their analysis, I was forced to adopt this strategy.

The second step involves discretization of numerical attributes. As the rough set theory is based on the concept of indiscernibility, numerical attributes should be discretized into intervals, so that numbers falling within the same interval are deemed as being indiscernible. The discretization intervals may be decided on manually or by using automatic algorithms.

If a data table is going to be used both for training and testing, it is very important that the test set is not used to decide upon the discretization intervals. Doing this would brake with the principle of not letting the test set be involved in any steps of the generation of a classifier. Because of this, the preprocessing step of discretization is done after the splitting of the table. The test set must be discretized according to the discretization found on the training set.

If the discretization is done manually without looking at the specific data to be analyzed, however, the discretization may be done before the table splitting. For instance the discretization may be done by an external expert on the domain under consideration using his domain knowledge, on for instance standard discretization intervals for the attributes.

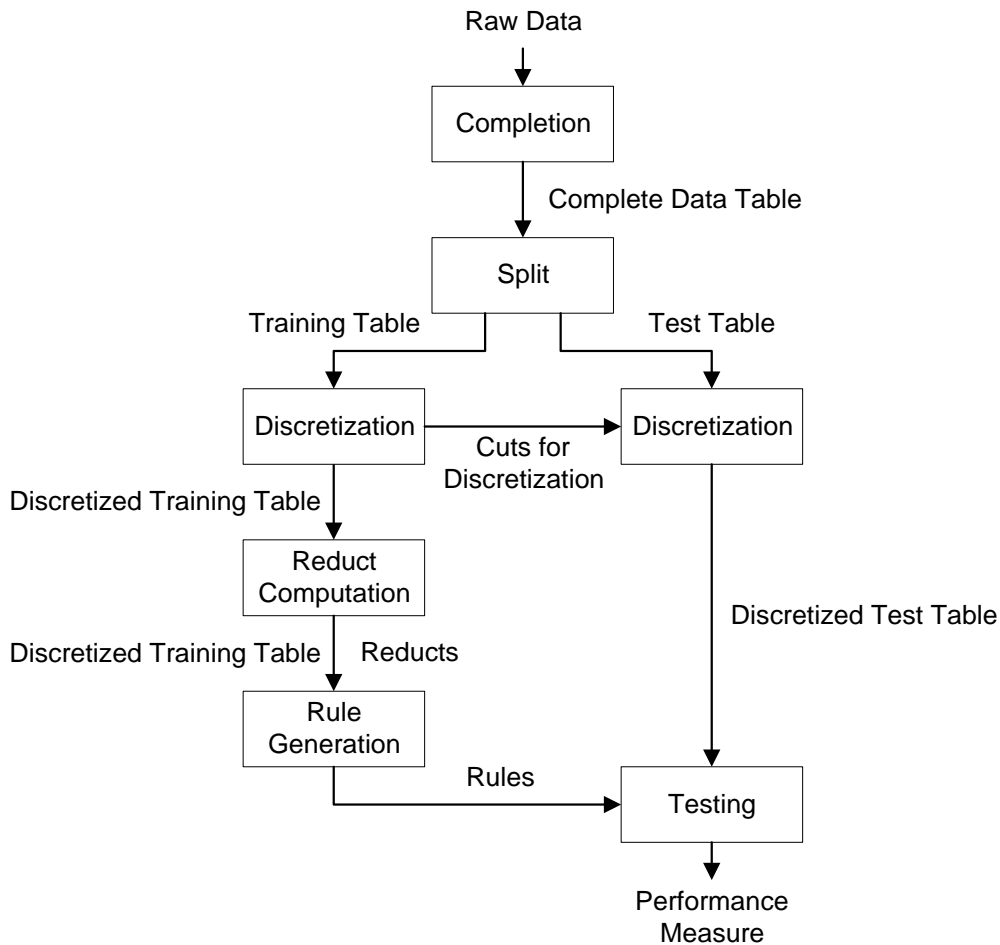


Figure 6: Training/testing cycle using the rough set approach

When using the rough set methodology, the first step in the training box of Figure 5 is to compute reducts (see Figure 6). This can be done with any type of reducts (see Section 2.2 and 2.3). After reduct computation, rules are generated from the reducts and the decision table. After the steps of reduct computation and rule generation, "weak" reducts and rules can be filtered. In the rough set approach, the classifier in Figure 5 is a set of rules, as can be seen in Figure 6. The objects in the test set are classified using the method described in Section 3.1 and the performance of the classifier is evaluated using some performance measure (see Section 3.3).

The analyses presented in Chapter 6 are performed in accordance with the steps described above. In the reduct computation process, different types of reducts has been computed on different variants of the original decision table: Full reducts, objected-related reducts, and full and object-related dynamic reducts with different sampling strategies. As the reducts has been computed with the aim of classification, reducts relative to the decision attribute has been computed in all of the analyses.

The sampling strategy is typically this: 2 fold cross-validation is performed on a data table 10 times with different splitting of the data table into training and test sets each with (approximately) equal size (approximately 50% of the original data table). This was done to make a good prediction of the performance of the method of generating the rough set classifier, by trying to reduce the effect of random variations resulting from any one particular splitting of the data or from the luck in finding the «best reducts».

The following performance parameters (average values) of the resulting 20 generated classifiers where recorded: Number of reducts, number of rules, sensitivity, specificity, accuracy, and the area under the ROC curve. The sensitivity, specificity, and accuracy measures where recorded using the voting strategy described in Section 2.5. The decision class with the largest number of votes were selected (decision threshold = 0.5). Using this decision threshold led to a little lower results for sensitivity, specificity, and accuracy than the «best» decision threshold. This is because the rough set models tended to overestimate the negative diagnosis. Since the negative diagnosis

was the majority decision, more rules were generated supporting this decision. This bias could have been compensated by using a lower prioritization threshold than 0.5 for the positive diagnosis.

All rough set computations were carried out using the ROSETTA software system (See Section 2.6).

6. Experiments

In this chapter different rough set models are presented. They are generated with different reduct computing algorithms (Johnson, genetic, or exhaustive), with different reduct types (object-related or full), different sampling strategies in the dynamic reduct computation, on different attribute subsets.

The notation for the sampling strategy for dynamic reduct computation in the table must be commented. If it says, for instance, 5, 10, 50, 90 in the column denoted Dyn. par. (meaning dynamic parameters), this should be interpreted as samplings on 5 levels with 10 different samples per level on sizes from 50% to 90% of the original table. The 5 levels are equally spaced.

One important objective in the analyses has been to investigate different filtering strategies for reducing the number of reducts and the size of the rule base. The results for these experiments have been presented as simple plots. The tables containing the background data have been put in the Appendix.

In section 6.3, 6.4, and 6.5 the logistic regression models by Hallan et al. on three different attribute subsets, respectively, is compared to the rough set models built using only the same attributes.

6.1 Experiment 1

In this experiment approximate algorithms for finding object-related reducts were tested. All attributes described in the Data Material section were included with the manual discretization of the numerical attributes. 2-fold cross-validation 10 times with different splittings of the data was carried out for each experiment. The results are shown in Table 8.

Exp.	Algorithm	No. Red.	Red. type	Dyn. par.	Reducts	Rules	AUC	SD
w1	Johnson	1	object-related	5, 10, 50, 90	372,95	610,70	0,8976	0,0320
r1	Genetic	1	object-related	5, 10, 50, 90	595,70	1087,55	0,9017	0,0258
r2	Genetic	2	object-related	5, 10, 50, 90	865,65	1921,10	0,9034	0,0252
r3	Genetic	3	object-related	5, 10, 50, 90	1107,20	2725,90	0,9030	0,0262
r5	Genetic	5	object-related	5, 10, 50, 90	1674,10	4410,35	0,9021	0,0254
r10	Genetic	10	object-related	5, 10, 50, 90	3362,95	9711,20	0,8979	0,0272
r1a	Genetic	1	object-related	5, 10, 10, 50	314,70	632,10	0,9113	0,0274
r1b	Genetic	1	object-related	5, 10, 10, 90	528,40	947,95	0,9099	0,0275

Table 8: Different approximate algorithms, all attributes

We see from the above table for experiments r1 to r10 that when the number of genetic reducts to search for is increased, the set of reducts and rules also get larger. The AUC is highest for experiment r2 and it decreases as more reducts are searched for. We also see that the experiment with the Johnson algorithm performs well, but lower than all the other experiments. Experiment r1a and r1b are the two best experiments in the table, even though they both have small reduct and rule sets. This is probably because the dynamic reducts are generated from small subtables.

As experiment r1a is the best, it will be studied further in the following sections. Different filtering strategies are tried out on the experiment. The data for the plots has been put in the Appendix.

6.1.1 Filtering on rule coverage

For the results presented in this section, the rule base was filtered on rule coverage. In Figure 7 and Figure 8 is the number of rules and the AUC plotted against the rule coverage filtering threshold, respectively.

Coverage is the fraction of objects in the decision table with the same decision as the rule that matches it. This means that coverage is equal to the rule support divided by the number of objects with the same decision as the

rule. The coverage measures the usualness of the rule. By taking into account the frequency of the decision, filtering on rule coverage is «kinder» to the minority decision than filtering on rule support. In the table, coverage = 0.05 means that all rules with coverage equal to 0.05 or lower were filtered away from the original rule set.

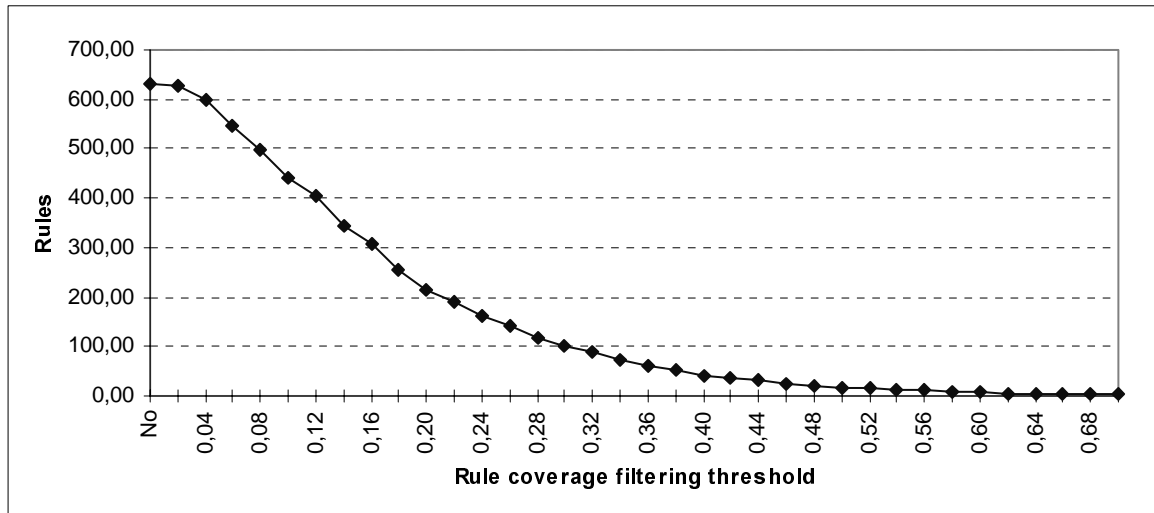


Figure 7: Filtering on rule coverage; Rules as a function of filtering threshold

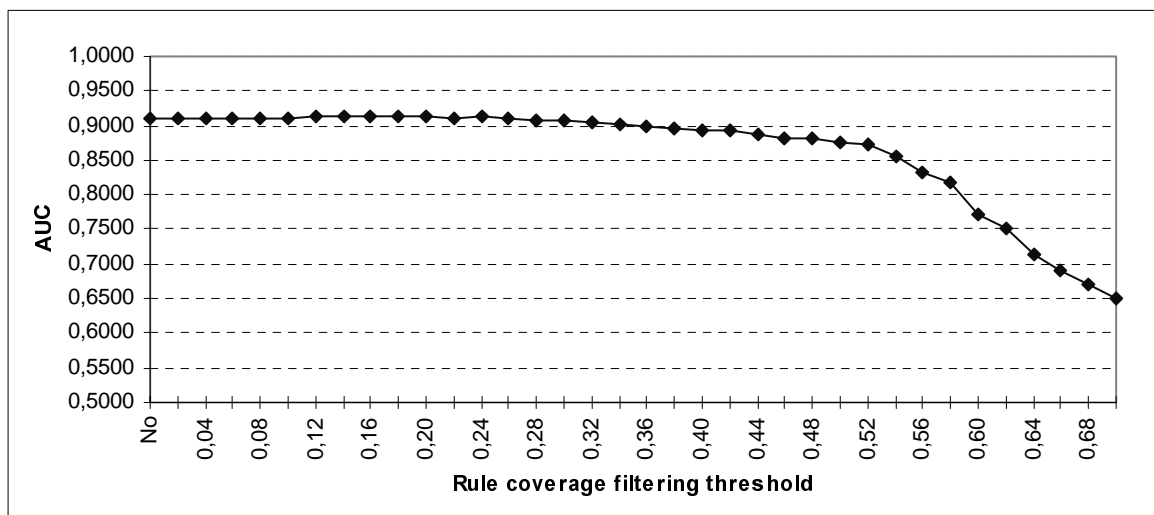


Figure 8: Filtering on rule coverage; AUC as a function of filtering threshold

We see from the figures that the AUC is very stable as the number of rules decreases for increasing value for the threshold. In fact, the AUC rises slightly in the beginning to a value of 0.9128 for coverage filtering threshold of 0.2. At this threshold the rule base has been reduced to one third of the original size. The AUC decreases then relatively slowly to a threshold of 0.52 after which the AUC decreases rapidly. At the threshold of 0.52, the AUC is 0.8734 and the rule base consists only of 16.1 rules.

6.1.2 Filtering on rule probability

In this section, the results when the rule base was filtered on rule probability is presented. In Figure 9 the number of rules and the AUC plotted against the rule probability filtering threshold, respectively.

The rule probability is given to each possible decision for a rule. For each decision it is equal to the fraction of the objects in the decision table with the same antecedent (rule's total support) that also has that particular decision. Filtering on the rule probability is the same as removing indeterminisms in the rule set (on a per rule basis). In the table, probability = 0.7 means that rules with probability less than 0.7 for both possible diagnoses is

removed from the set of rules. This means that rule 1 would be filtered and rule 2 would not be filtered in the following example:

rule 1: $a1=v1$ and $a2=v2 \rightarrow d = \text{yes}$ (probability 0.6) $d = \text{no}$ (probability 0.4)
 rule 2: $a1=v3$ and $a2=v4 \rightarrow d = \text{yes}$ (probability 0.2) $d = \text{no}$ (probability 0.8)

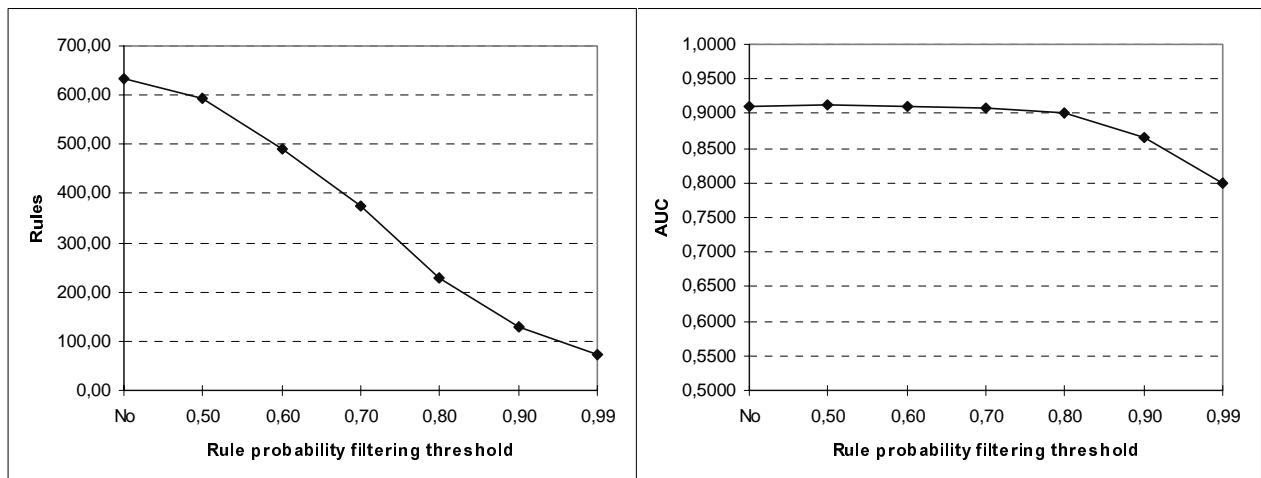


Figure 9: Filtering on rule probability; Rules and AUC as a function of filtering threshold

When filtering on probability, we see from Figure 9 that the AUC is reasonably stable to a filtering threshold of 0.76. At this threshold the rule base has been reduced to a little over 1/3. The top point on the AUC plot is for threshold 0.5 where the AUC is 0.9117, 0.004 above the AUC for no filtering (which is insignificant). The rule base has only been slightly reduced with this threshold.

6.1.3 Filtering on rule stability

Here, the results when the rule base was filtered on rule stability is presented, see Figure 10.

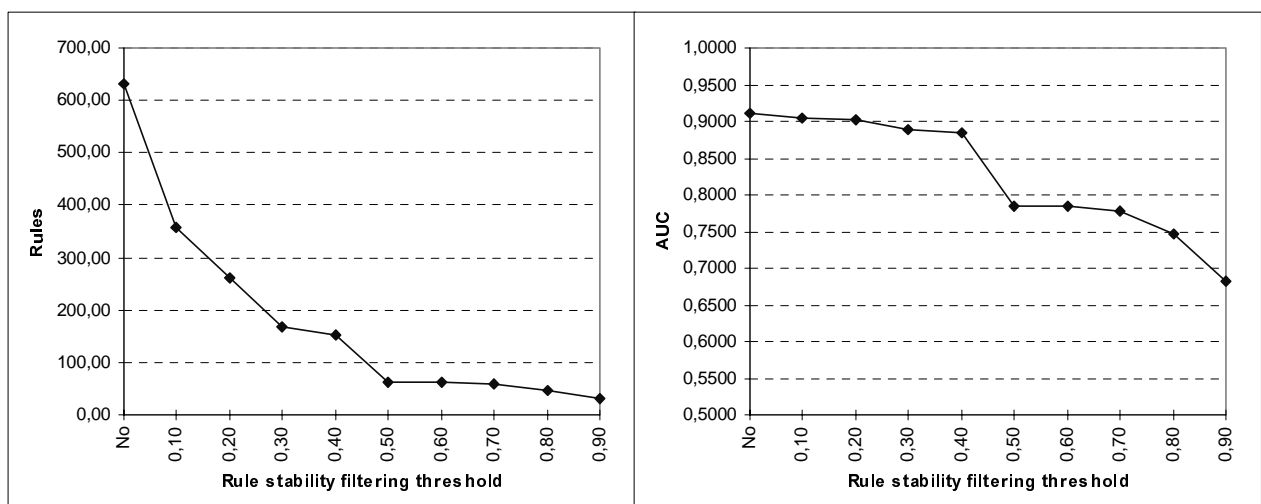


Figure 10: Filtering on rule stability; Rules and AUC as a function of filtering threshold

We see from the figure that the AUC plot is monotonically decreasing. The decrease is relatively low to a threshold of 0.4. At this threshold the rule base is approximately 1/4 of the original rule base with no filtering. After this threshold, there is a relatively large decrease in the rule base which results in AUC values below 0.8.

6.1.4 Filtering on reduct length

In this section we will look at the effect of filtering on reduct length. Filtering on reduct length is motivated by the assumption that short rules are more general than long rules (Occam's razor). The rule base consisted only of reducts with a length of 1, 2, 3, and 4. The possible filtering thresholds are then the ones shown in Figure 11.

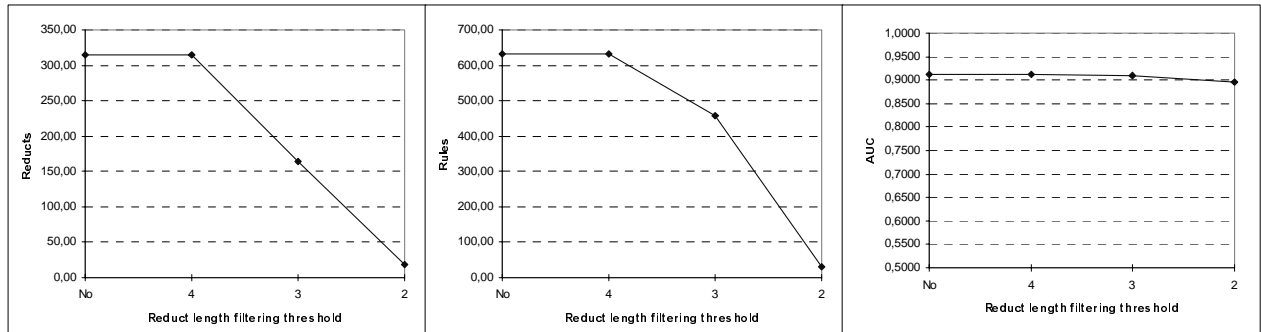


Figure 11: Filtering on reduct length; Reducts, rules and AUC as a function of filtering threshold

We see that rules with left hand side lengths of 1 and 2, classifies almost as well as all the rules in the original rule base. This is not very surprising, as there are most rules with length 2. When only rules with left hand side length of 1 is used, the AUC drops only to 0.8969, even though the size of the rule base is reduced significantly to only 31.2 rules (on average). The number of reducts with length 1 is 18.50 on average, which means that most of the attributes are found as reducts, as there are 19 attributes in the analysis. This means that a simple strategy of generating rules from a set of reducts where each attribute is a «reduct» performs very well on the current data set. This is a strategy that should be tested when a new data set is analysed.

6.1.5 Filtering on reduct support

Filtering on reduct support is the same kind of «filtering» as in (F, ϵ) -generalized dynamic reducts, where $\epsilon = \text{reduct support} / \text{number of subtables}$. If the number of subtables in the dynamic reduction process is equal to n , filtering on reduct stability ϵ is the same as filtering on reduct support $n \cdot \epsilon$. In this case $n = 50$.

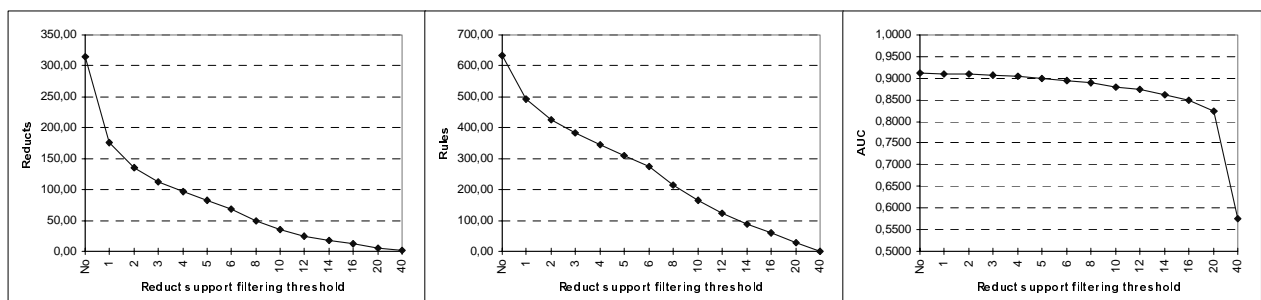


Figure 12: Filtering on reduct support; Reducts, rules and AUC as a function of filtering threshold

Note that in Figure 12 there are not equal steps between the thresholds.

We see from the figure that the AUC is decreasing as the threshold is increased. The largest decrease in the reduct and rule sets is for the first threshold. A threshold of 4 gives an AUC of 0.9041, 96.90 reducts, and 346.35 rules.

6.1.6 Filtering by removing single attributes

In this section reducts containing certain attributes will be removed (one at a time).

When removing reducts containing single attributes from the reduct set, we hope to find out which attributes are absolutely necessary to perform a good classification. The results of filtering by removing reducts containing single attributes is shown in Figure 13. The AUC when no attributes had been removed is also shown in the figure.

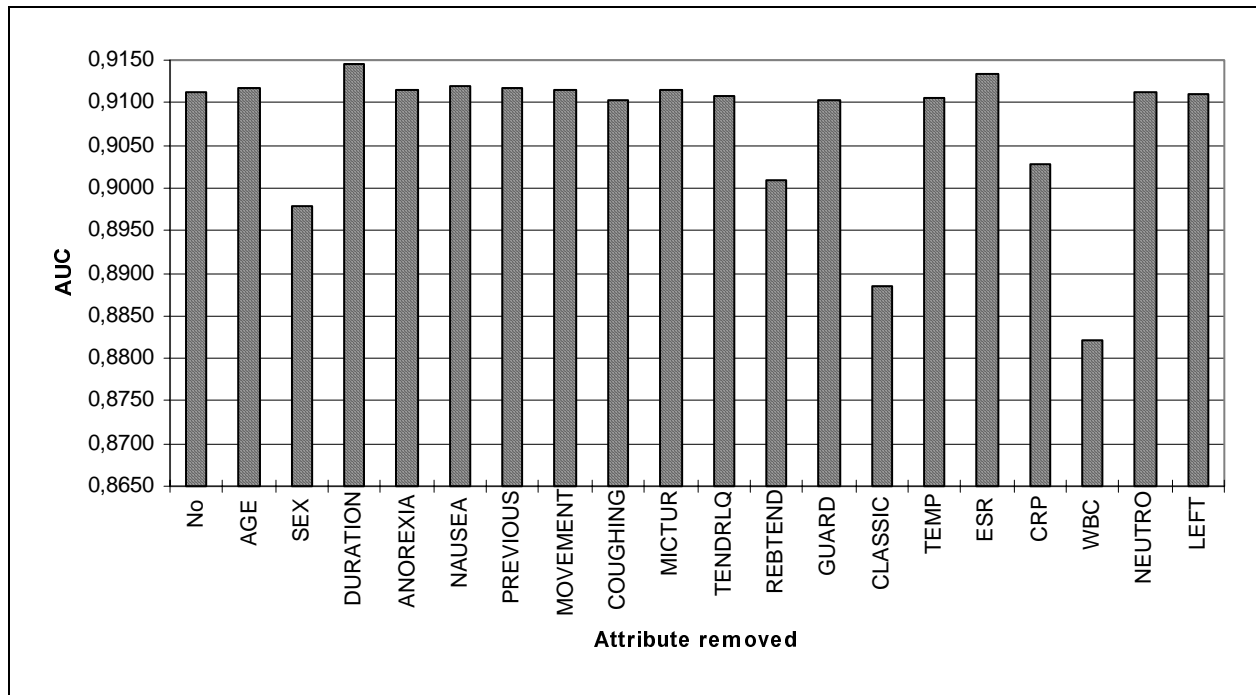


Figure 13: Filtering by removing single attributes; AUC as a function of filtering attribute

The attributes may now be ranked according to the AUC.

Filtering by removing attributes seems to be very successful. It is obvious that certain attributes confuses the rough set models. On the other hand, there are also some attributes that should not be removed from the analysis.

More attributes could be removed in order to try to find the «optimal» attribute subset to be subjected to rough set analysis. For instance one might remove all attributes that led to an AUC larger than the original when the reducts containing them were removed.

6.1.7 Comparison

In this section, the different filtering strategies are going to be compared. The AUC is plotted against the number of rules for the different filtering strategies. The highest curve can be said to be the best, but one has to remember that the filtering strategies do influence on the structure of the rules in the rule base as well as the number of rules. For instance may short rules be preferred to long rules. This is not shown directly in the plot.

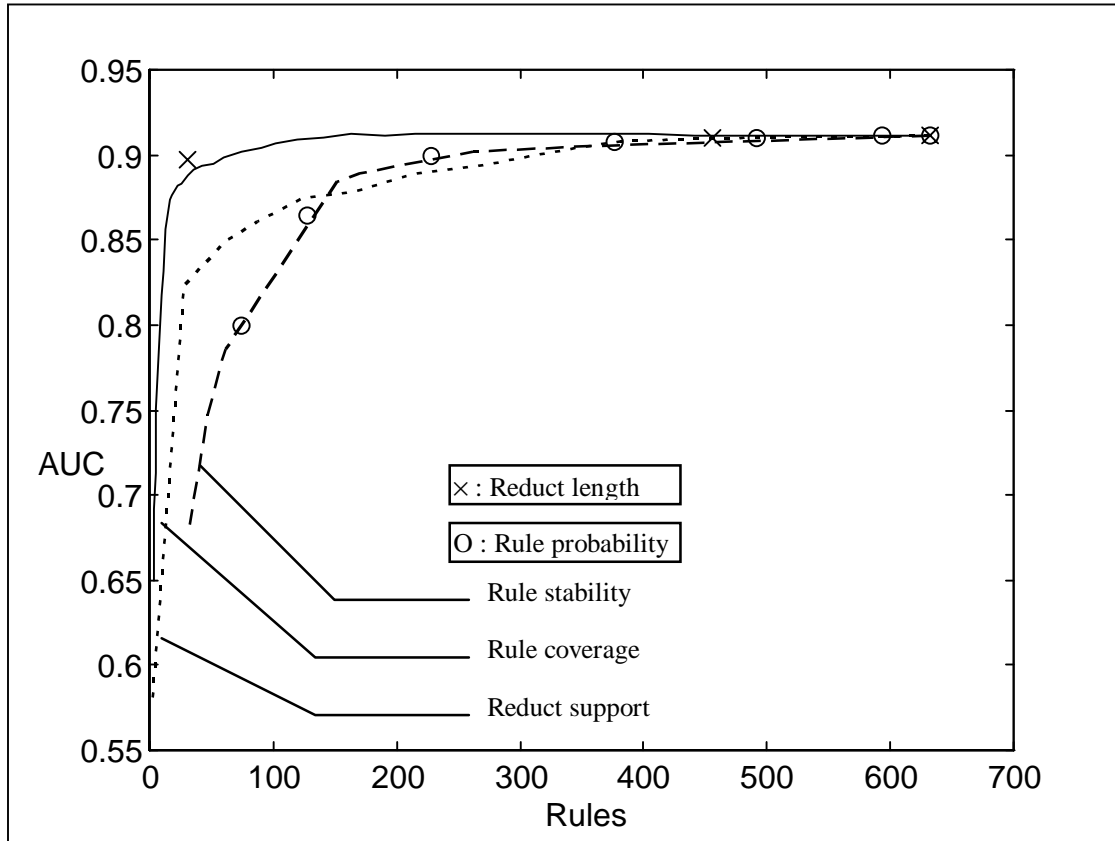


Figure 14: Comparison of the filtering strategies; AUC as a function of number of rules

Filtering on rule support seems to be better than the other filtering strategies, as it has higher AUC values than most of the other strategies for small rule bases. We see that an «x» has come above the solid rule coverage line. This means that filtering on reduct length is better for one particular rule base size (the one represented by the point). As the rule coverage has much more filtering values than the reduct length, one may choose to use a slightly higher rule base that gives an higher AUC than the «x» above the solid line.

6.2 Experiment 2

In this section similar experiments as in the last section (section 6.1) will be done using full reduct computation instead of object-related reduct computation.

Exp.	Algorithm	No. Red.	Red. type	Dyn. par.	Reducts	Rules	AUC	SD
u1	Genetic	10	full	5, 10, 50, 90	448,80	43050,45	0,8756	0,0279
u2	Genetic	50	full	5, 10, 50, 90	405,45	43018,95	0,8665	0,0278
u3	Genetic	100	full	5, 10, 50, 90	493,20	52896,80	0,8673	0,0270
u4	Genetic	10	full	5, 10, 10, 50	534,75	27422,60	0,9109	0,0280
u5	Genetic	10	full	5, 10, 10, 90	512,25	35583,50	0,9113	0,0295

Table 9: Genetic algorithm; All attributes

We see from Figure 14 that when the number of reducts searched for by the genetic algorithm from 10 to 100, the reduct and rule bases increases slightly, while the AUC decrease slightly. Just like for object-related reducts, dynamic reduct sampling of small subtables, leads to better results. The higher number of reducts but lower number of rules indicates that shorter reducts has been found.

In the following sections the best result (u5) is analysed further by using different filtering strategies on both reducts and rules. The data for the plots have been put in the Appendix.

6.2.1 Filtering on rule coverage

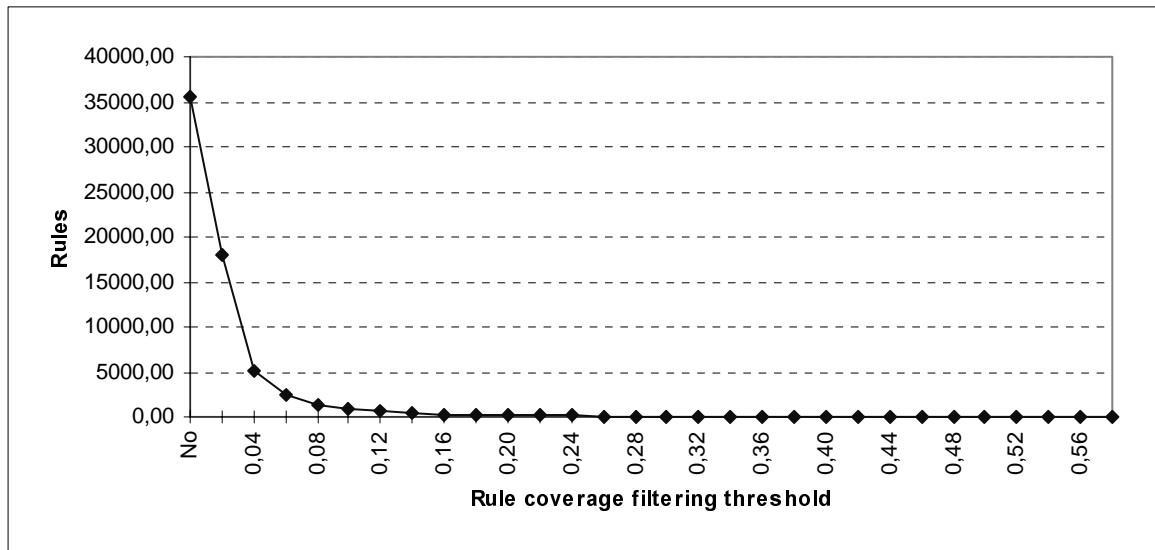


Figure 15: Filtering on rule coverage; Rules as a function of filtering threshold

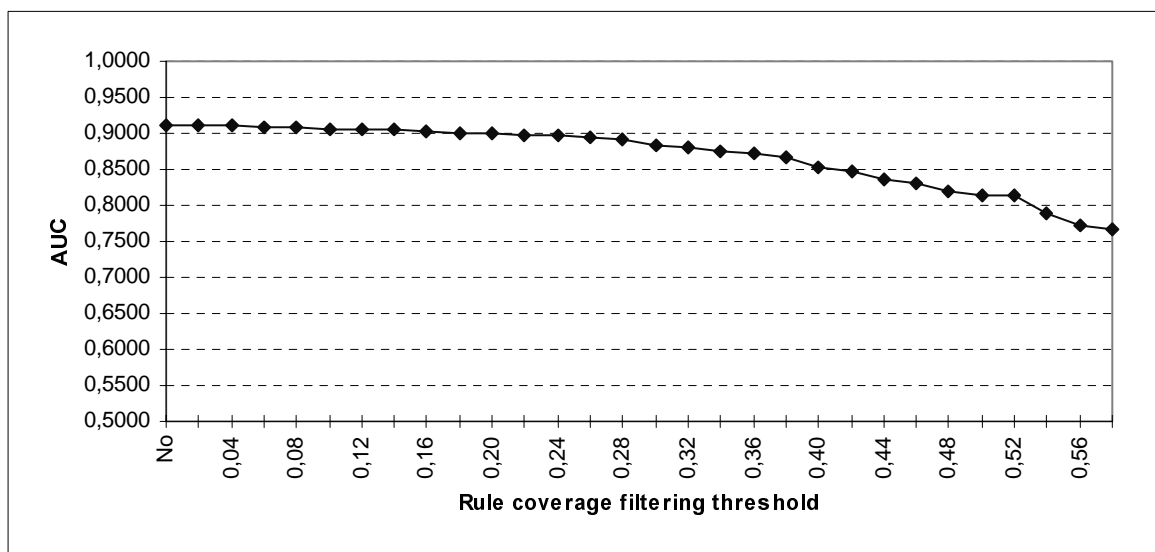


Figure 16: Filtering on rule coverage; AUC as a function of filtering threshold

We see that for the first filtering values, the rule base is reduced very much. At a filtering threshold of 0.16, the AUC has been reduced with only ca 0.008 to 0.9034, while the rule base has been reduced from approximately 35 500 to 322.3 rules. To see more details, the figures have been redrawn in Figure 17 and Figure 18, respectively, now with threshold values from 0.1 to 0.58.

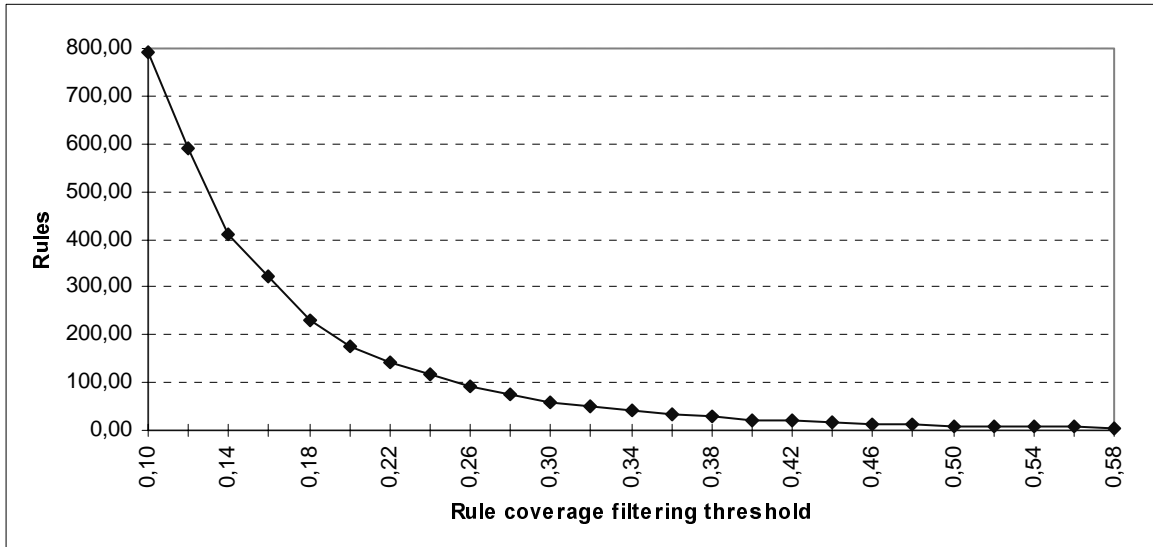


Figure 17: Filtering on rule coverage; Rules as a function of filtering threshold

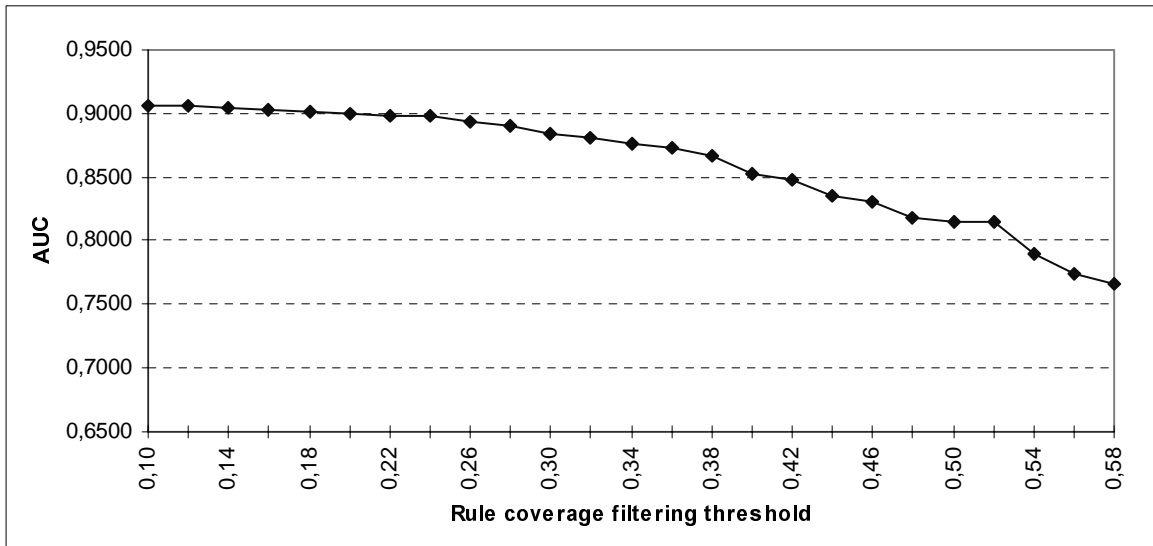


Figure 18: Filtering on rule coverage; Reducts, rules and AUC as a function of filtering threshold

6.2.2 Filtering on rule probability

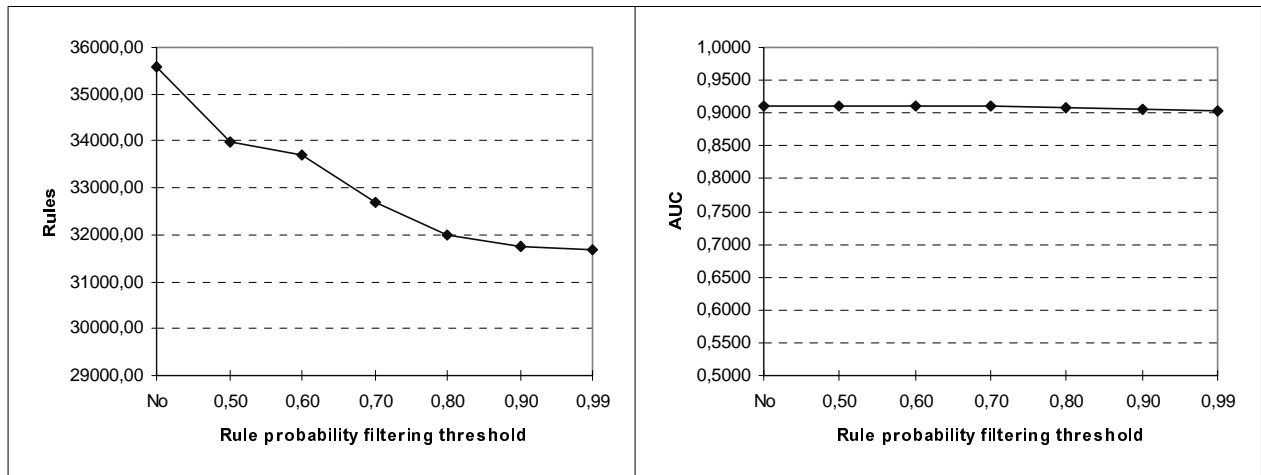


Figure 19: Filtering on rule probability; Rules and AUC as a function of filtering threshold

Filtering on rule probability reduces the rule base only from ca 36 000 to 32 000 rules, with a minimal decrease in the AUC. The rule base is at its minimum for the threshold 0,99. At this threshold, the rule base only consists of consistent rules. This means that the rules generated with full dynamic reducts are typically more consistent than when the rules are generated with object-related reducts (see section 6.1.2).

6.2.3 Filtering on reduct length

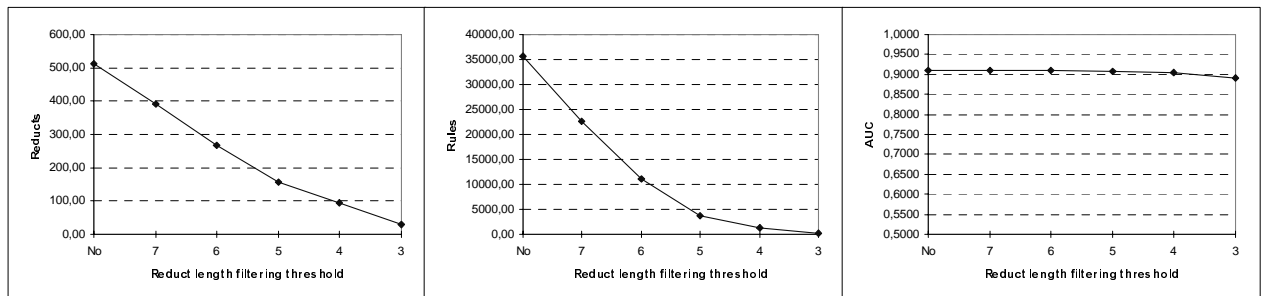


Figure 20: Filtering on reduct length; Reducts, rules and AUC as a function of filtering threshold

The number of reducts and rules decreases rapidly when the rule base is filtered on reduct length, which indicates that there are many long reducts (and rules) in the original rule base. We see that the number of rules decreases even more rapid than the number of reducts. This reflects the fact that there typically are generated more rules from long reducts than short reducts. This is not surprising, as the longer the rule is, the more specialized it gets. Then there are more possibilities for generating rules.

We see that there is not very interesting to remove inconsistencies in a rule base generated from full dynamic reducts, as it does not contain that many inconsistencies. One typically gets longer and more specialized rules when full dynamic reducts are computed than when object-related rules are computed.

6.2.4 Filtering on reduct support

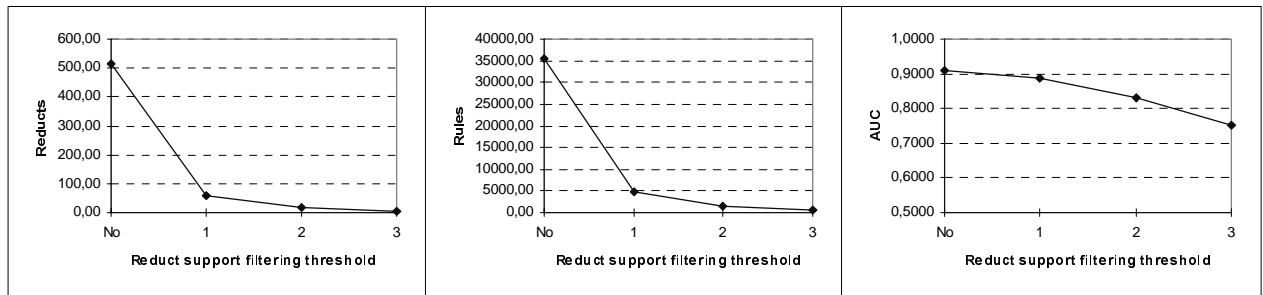


Figure 21: Filtering on reduct support; Reducts, rules and AUC as a function of filtering threshold

The reducts generated by full dynamic reducts only had support counts in the range [1, 3] in at least one of the 20 executions. Therefore, only three different filtering thresholds were examined. We see from the figure that we get an acceptable AUC when reducts with support count 1 were filtered. The rule base consisted then of approximately 5 000 rules.

6.2.5 Filtering by removing single attributes

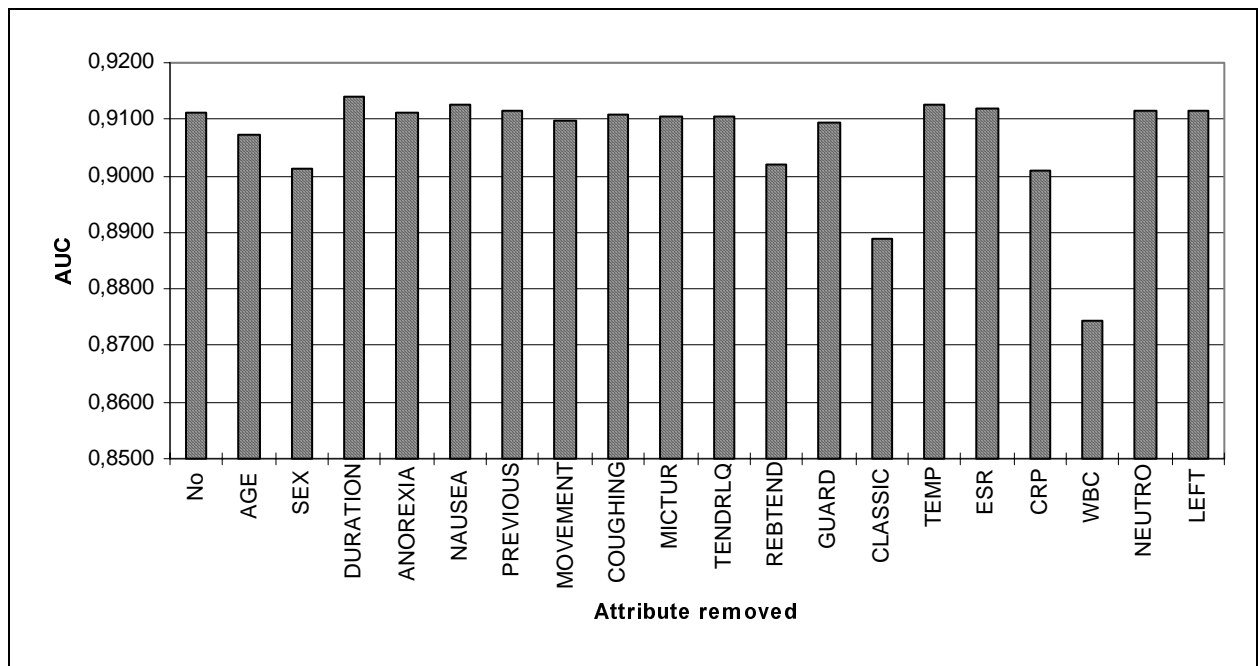


Figure 22: Filtering by removing single attributes; AUC as a function of filtering attribute

6.3 Experiment 3

In this section one of the attribute subsets studied by Hallan et al. is analyzed. The attribute subset, called A for simplicity consisted of the following clinical attributes: CLASSIC, REBTEND, SEX, TENDRLQ, COUGHING, GUARD. The results from the logistic regression model by Hallan et al. is shown in the last row but one. In the last row the performance of the surgeons is shown. This is a fair comparison as the only information available for all «models» were clinical variables.

Exp.	Algorithm	Red. type	Dyn. par.	Reducts	Rules	AUC	SD
x1	Exhaustive	full	10, 50, 5, 50	42,00	486,85	0,8457	0,0236
x2	Exhaustive	full	10, 50, 50, 95	2,30	66,75	0,8077	0,0297
x3	Exhaustive	full	10, 50, 5, 95	54,35	522,25	0,8495	0,0263
Log. regression	-	-	-	-	-	0,854	0,0283
Surgeons	-	-	-	-	-	0,817	-

Table 10: Comparison of models built on attribute subset A

We see from the table that the logistic regression performs best. Of the rough set analyses, experiment x3 gives the highest AUC. The best rough set model is better than the estimates done by the surgeons.

In the following sections experiment x3 is analysed by using different filtering strategies. The data for some of the plots has been put in the Appendix.

6.3.1 Filtering on rule coverage

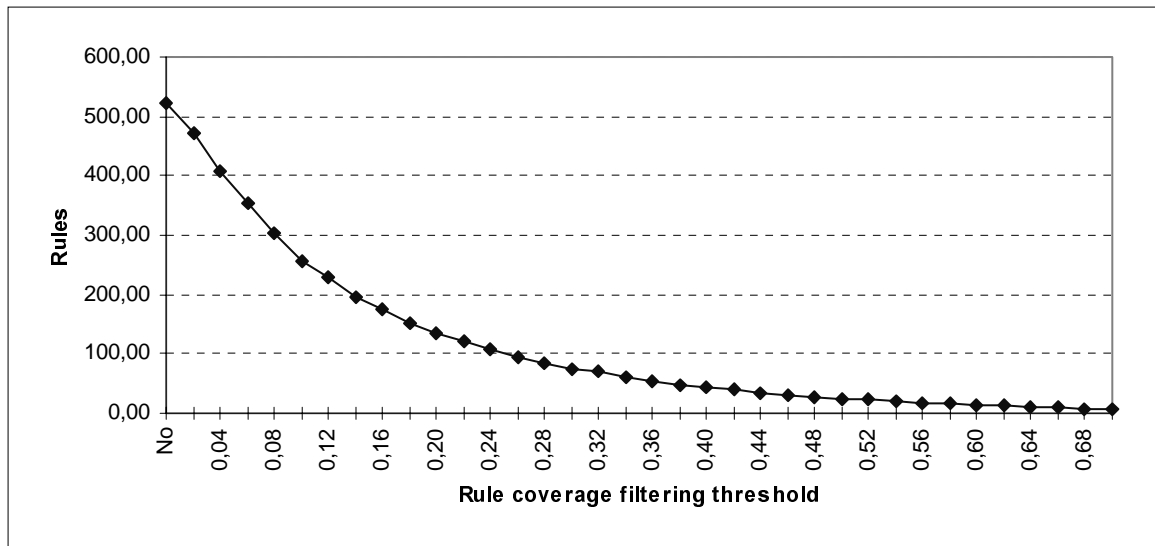


Figure 23: Filtering on rule coverage; Rules and AUC as a function of filtering threshold

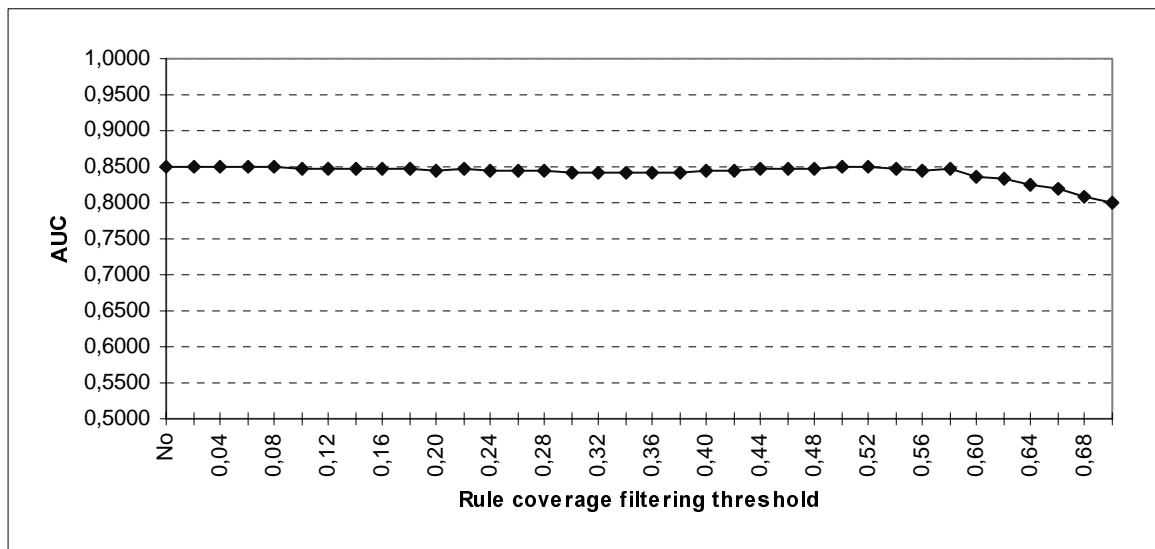


Figure 24: Filtering on rule coverage; AUC as a function of filtering threshold

We see that the AUC first decreases slightly and then increases slightly. With thresholds higher than 0.58, the AUC decreases slowly. For a better view of details, the figures has been redrawn in Figure 25 and Figure 26, respectively, now with threshold values from 0.38 to 0.58.

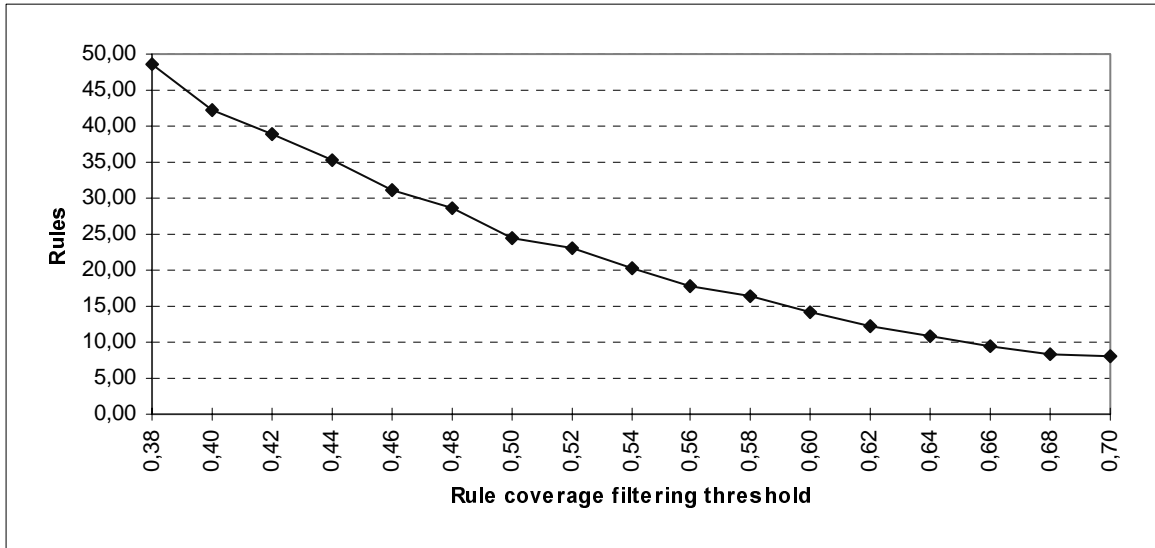


Figure 25: Filtering on rule coverage; Rules as a function of filtering threshold

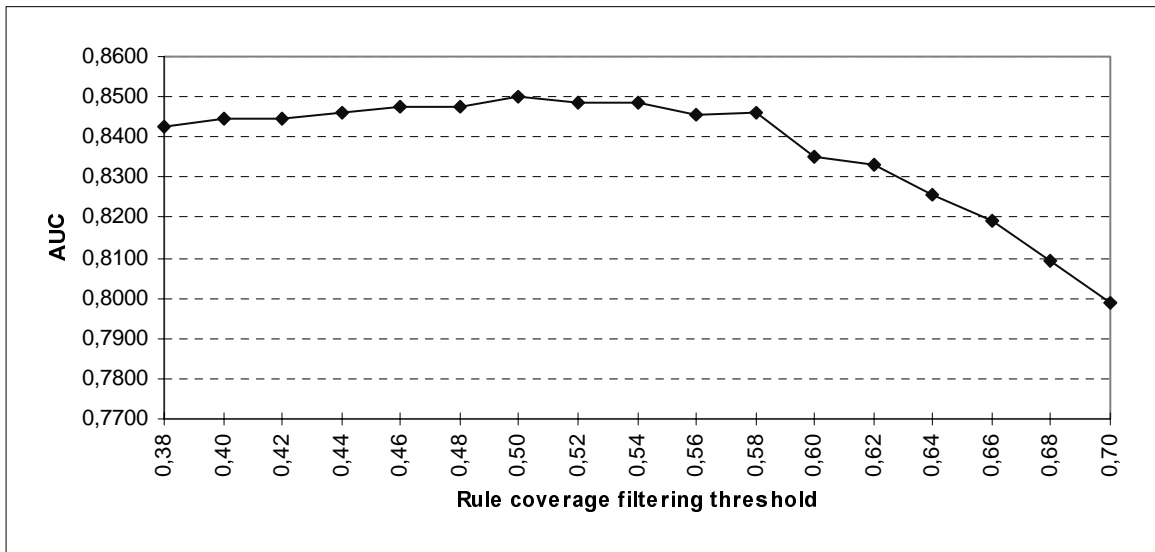


Figure 26: Filtering on rule coverage; AUC as a function of filtering threshold

We see from Figure 26 that a coverage threshold of 0.50 gives the best AUC value of 0.8502, slightly but insignificantly better than with no filtering on rule coverage. The gain lies in the reduction of the rule base. The rule base consists of only 24.40 rules, that is ca 1/20 of the unfiltered rule base. For a threshold of 0.58, there are only 16.45 rules, but an AUC of 0.8462.

6.3.2 Filtering on rule probability

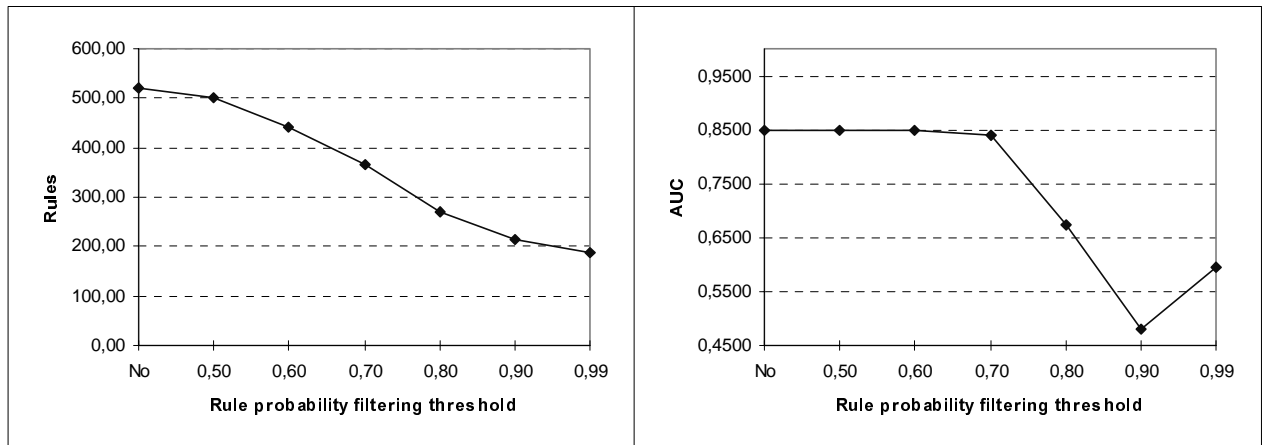


Figure 27: Filtering on rule probability; Rules and AUC as a function of filtering threshold

We see clearly from the above figure that the AUC is almost equal for thresholds 0.50, 0.60, and 0.70. For higher filtering thresholds there is a distinct drop in the AUC. For a threshold of 0.70, the rule base has been reduced from 522.25 (for no filtering) to 365.85 rules. The AUC has been reduced from 0.8495 to 0.8391.

6.3.3 Filtering on reduct length

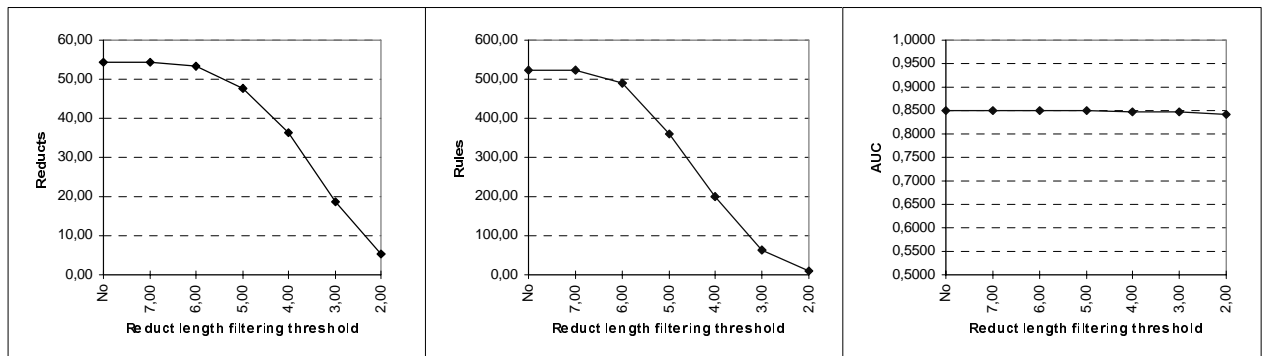


Figure 28: Filtering on reduct length; Reducts, rules and AUC as a function of filtering threshold

From the above plots in Figure 28 we can see that filtering on reduct length does almost not imply any loss in the AUC. The same effect appeared in section 6.2.3. The number of reducts is (on average) 5.20. The number of rules is 10.40, which means that each reduct (of length 1) has two associated rules.

6.3.4 Filtering on reduct support

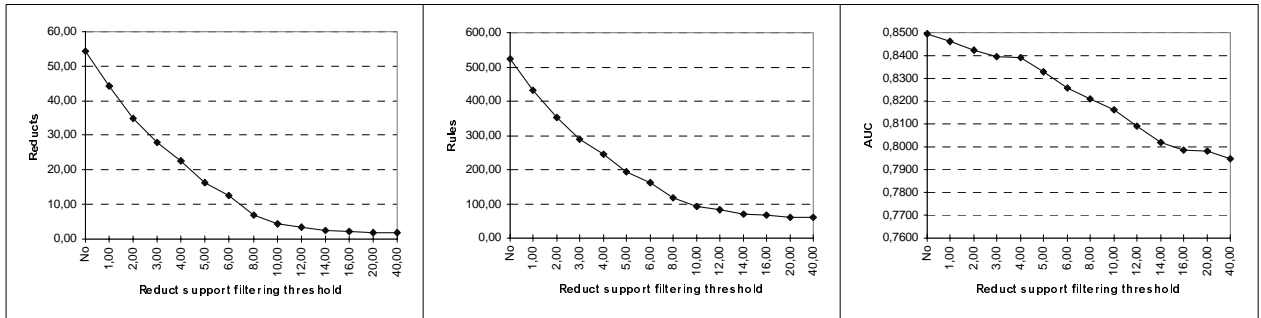


Figure 29: Filtering on reduct support; Reducts, rules and AUC as a function of filtering threshold

We see from Figure 29 that The AUC decreases slowly from as the reduct support threshold is increased. At a threshold of 4, the AUC is 0.8391, ca 0.01 lower than with no filtering. The set of reducts has simultaneously been reduced to ca 2/5 of the original reduct set, and the rule base has been reduced to ca 1/3.

6.3.5 Filtering by removing single attributes

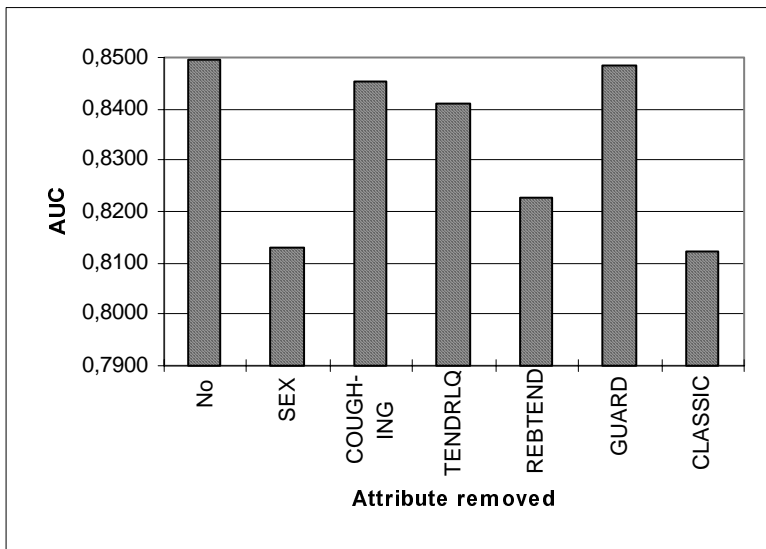


Figure 30: Filtering by removing single attributes; AUC as a function of filtering attribute

We see that filtering by removing reducts containing single attributes does not increase the AUC. The column chart in Figure 30 shows the relative importance of the attributes, however. The AUC can be treated as some kind of a score indicating the «goodness» of the attributes. CLASSIC can then be ranked as the best attribute followed by SEX, REBTEND, TENDRLQ, COUGHING, and GUARD.

6.4 Experiment 4

This section presents results from the analysis of the second attribute set studied by Hallan et al. This rule set, called B for simplicity, has the attribute WBC in addition to the attributes analysed in the preceding section. Thus, it consists of the attributes CLASSIC, REBTEND, SEX, TENDRLQ, COUGHING, GUARD, and WBC.

Exp.	Algorithm	Red. type	Dyn. par.	Reducts	Rules	AUC	SD
x1	Exhaustive	full	10, 50, 5, 50	96,55	1809,20	0,9087	0,0248
x2	Exhaustive	full	10, 50, 50, 95	5,25	254,85	0,8610	0,0389
x3	Exhaustive	full	10, 50, 5, 95	105,15	1744,50	0,9061	0,0250
Log. regression	-	-	-	-	-	0,901	0,0174

Table 11: Comparison of models built on attribute subset B

We see that two of the rough set models are slightly better than the logistic regression model, while the logistic regression model has a lower SD value.

In the following sections the result from experiment x3 is analysed by using different filtering strategies. The data for some of the plots has been put in the Appendix.

6.4.1 Filtering on rule coverage

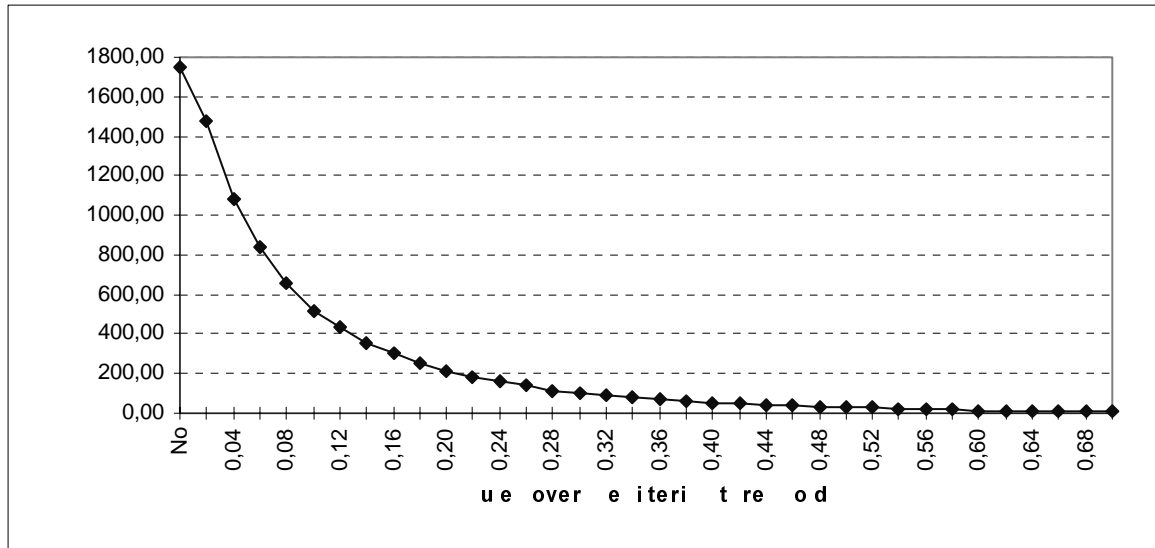


Figure 31: Filtering on rule coverage; Rules as a function of filtering threshold

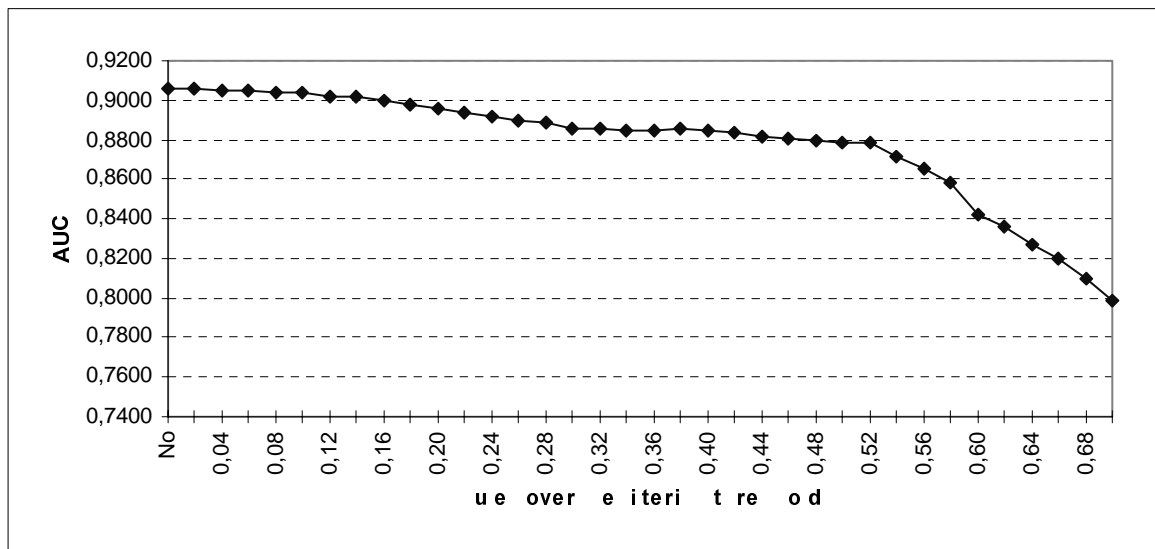


Figure 32: Filtering on rule coverage; AUC as a function of filtering threshold

We see from Figure 32 that the AUC decreases monotonically for increasing values of the threshold, in contrast to what happened in section 6.3.1 for the same analysis on attribute subset A. This attribute subset has attribute WBC in addition. For thresholds lower than 0.52 the decrease in AUC is relatively slow, but it is faster for higher thresholds. Still, we can see from Figure 31 that the rule base decreases very fast.

6.4.2 Filtering on rule probability

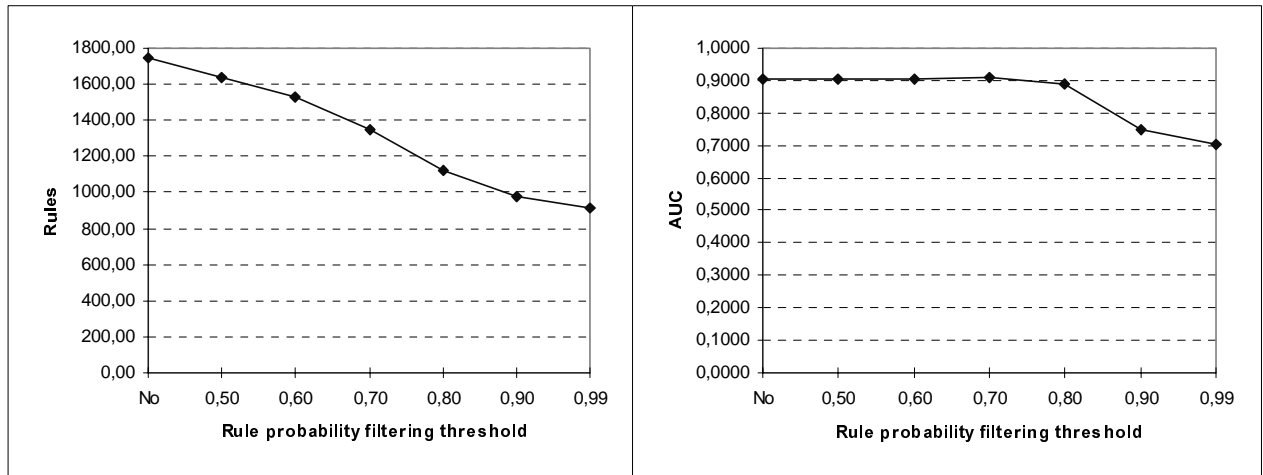


Figure 33: Filtering on rule probability; Rules and AUC as a function of filtering threshold

When filtering on rule probability, the AUC is fairly stable and actually slowly increasing when the threshold is increased from no filtering to filtering on rule probability of 0.70. The decrease in the rule base is not very fast, however. For a threshold of 0.70, the AUC is 0.9086. This is only 0.0025 higher than the AUC for no filtering. The rule base has decreased from 1744.50 to 1344.75, which is not much.

6.4.3 Filtering on reduct length

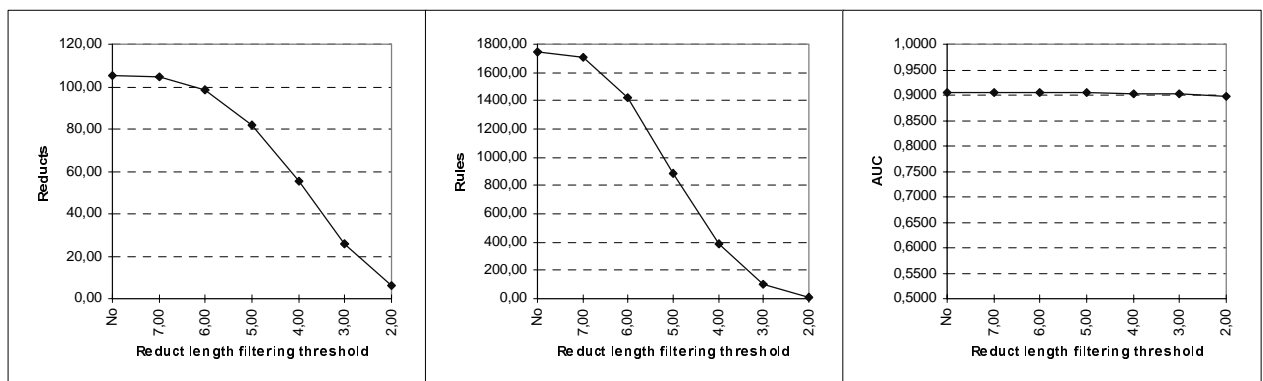


Figure 34: Filtering on reduct length; Reducts, rules and AUC as a function of filtering threshold

Again we see the effect commented in section 6.2.3 and 6.3.3. Short rules seem to classify almost as well as a large collection of different rule lengths, including the short rules. 6.2 (on average) out of 7 possible reducts of length 1 were found.

6.4.4 Filtering on reduct support

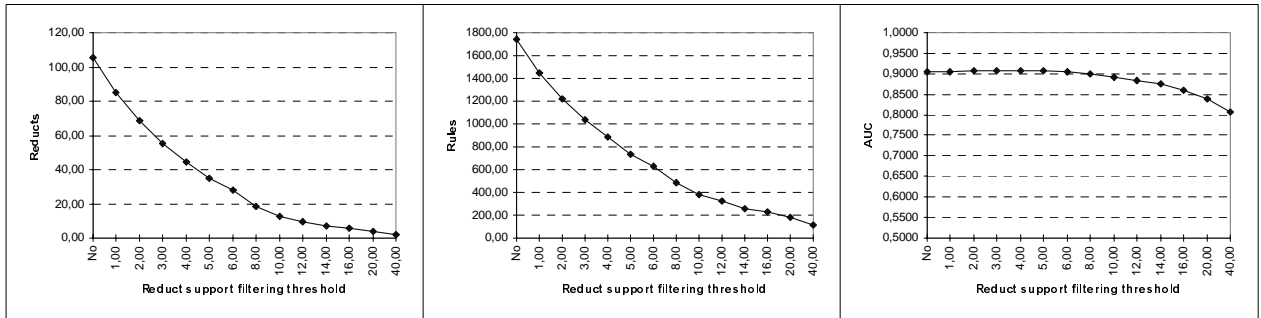


Figure 35: Filtering on reduct support; Reducts, rules and AUC as a function of filtering threshold

In contrast to in section 6.3.4, filtering on reduct support is successful when only the attribute WBC is included in the analysis in addition. We get a slow increase in the AUC to a top point at a filtering threshold of 4. After this, there is a slow decrease. At the threshold of 4, the AUC is 0.9086, while the AUC with no filtering is 0.9061. The rule base has been reduced to ca 1/2.

6.4.5 Filtering by removing single attributes

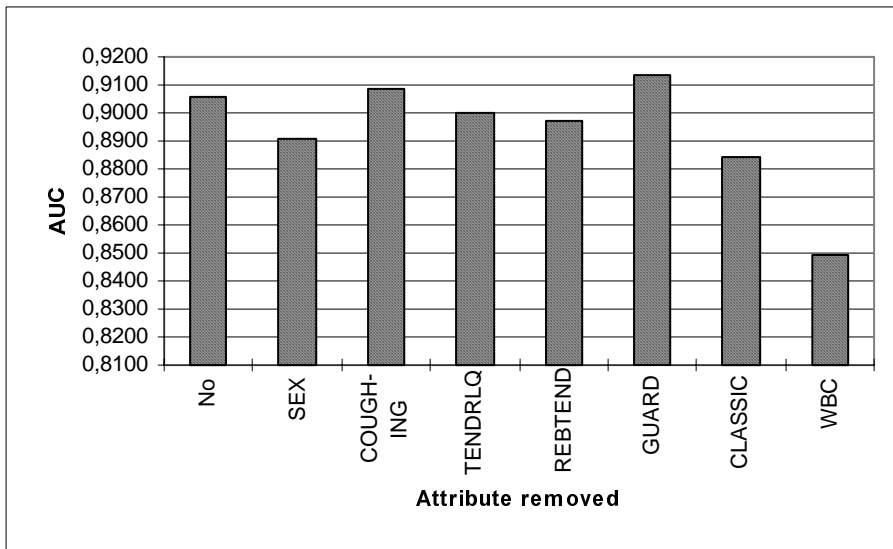


Figure 36: Filtering by removing single attributes; AUC as a function of filtering attribute

From Figure 36 we see that we get a distinct increase in the AUC when all reducts containing GUARD are removed from the attribute subset. There is also an increase for COUGHING. These are the least interesting attributes in the rough set models. We see that the WBC is the «best» attribute, followed by CLASSIC, SEX, REBTEND, TENDRLQ, COUGHING, and GUARD. This is almost the same ranking as in section 6.3.5.

6.5 Experiment 5

In this section results from the analysis of the third attribute set studied by Hallan et al. is presented. The attribute subset, called C for simplicity, consists of the attributes CRP and NEUTRO in addition to the attributes in the set B analyzed in the preceding section. C consists of the following attributes: CLASSIC, REBTEND, SEX, TENDRLQ, COUGHING, GUARD, CRP, WBC, and NEUTRO.

Exp.	Algorithm	Red. type	Dyn. par.	Reducts	Rules	AUC	SD
x1	Exhaustive	full	10, 50, 5, 50	433,80	16379,25	0,9246	0,0246
x2	Exhaustive	full	10, 50, 50, 95	59,80	4620,35	0,8877	0,0173
x3	Exhaustive	full	10, 50, 5, 95	423,50	15483,85	0,9249	0,0245
Log. regression	-	-	-	-	-	0,920	0,0238

Table 12 Comparison of models built on attribute subset c

We see that two of the rough set models are slightly better than the logistic regression model. The best experiment, x3, has an AUC of 0.9249, which is approximately 0.0049 higher than the logistic regression model.

The experiment giving the highest AUC is studied further in the following sections. It is analysed using different filtering strategies. The data for the plots have been put in the Appendix.

6.5.1 Filtering on rule coverage

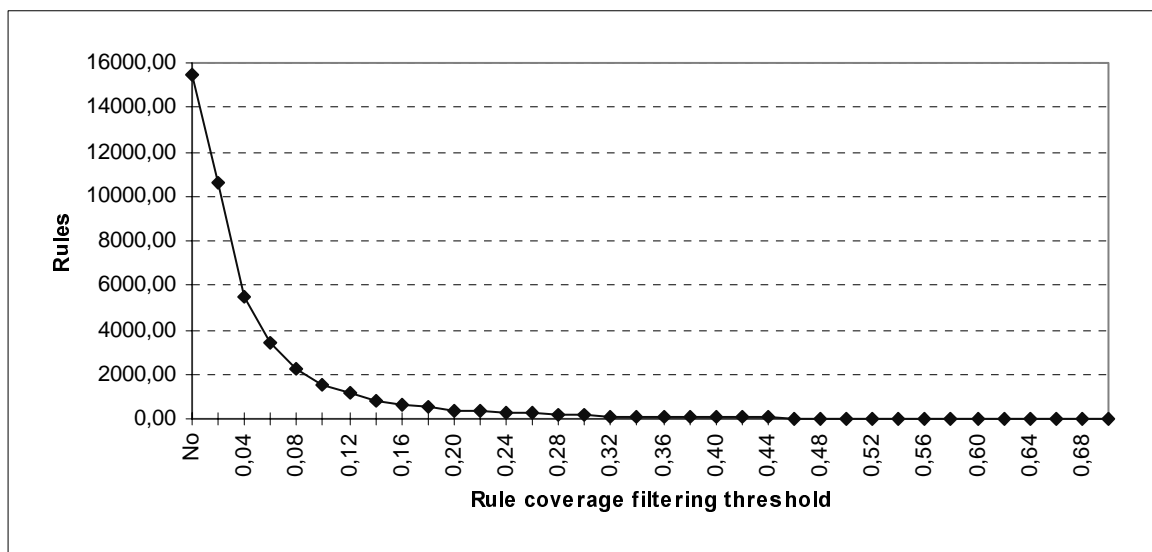


Figure 37: Filtering on rule coverage; Rules as a function of filtering threshold

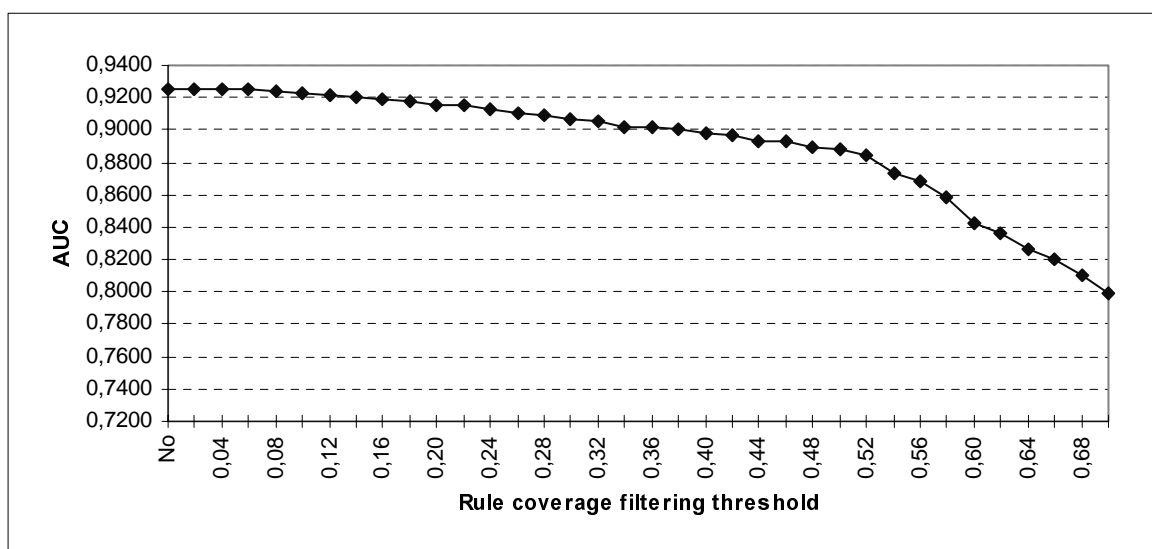


Figure 38: Filtering on rule coverage; AUC as a function of filtering threshold

First the AUC increases slightly to a threshold of 0.04. Then the AUC decreases slowly to a threshold of 0.52. After this threshold the AUC decreases somewhat faster. The size of the rule base decreases very rapidly, especially in the beginning.

6.5.2 Filtering on rule probability

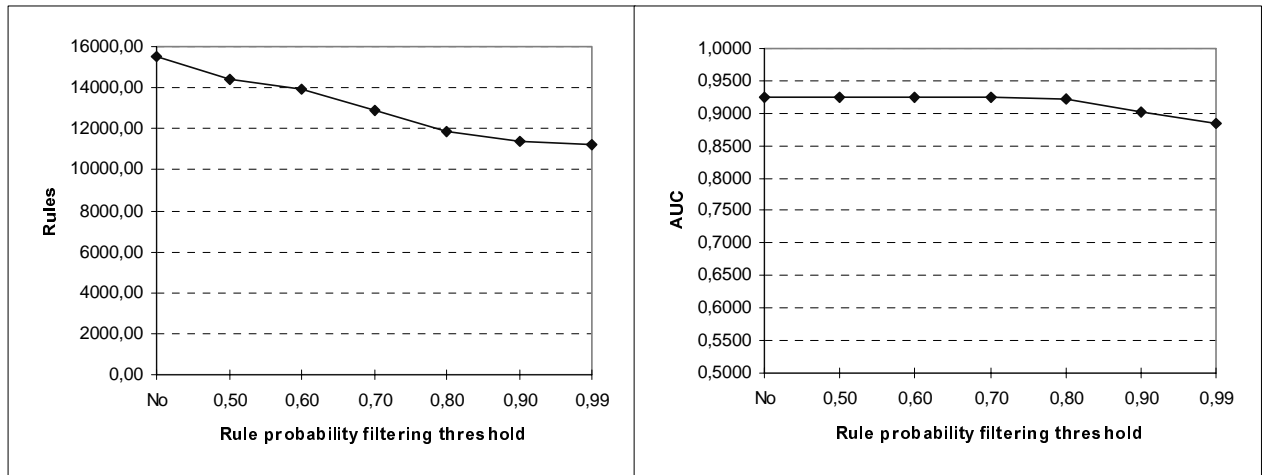


Figure 39: Filtering on rule probability; Rules and AUC as a function of filtering threshold

The AUC increases slightly when the threshold is increased from 0.50 to 0.80. Again we see that filtering on rule probability reduces the number of reducts only slightly.

6.5.3 Filtering on reduct length

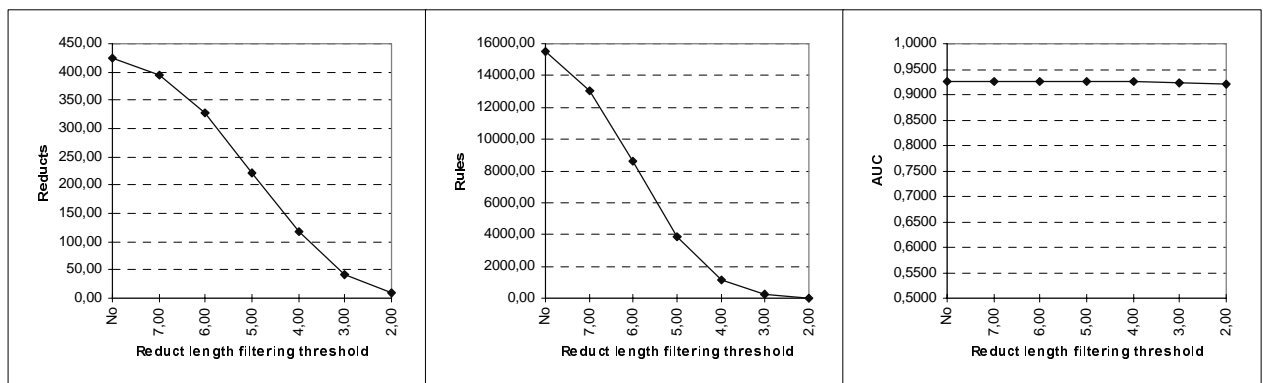


Figure 40: Filtering on reduct length; Reducts, rules and AUC as a function of filtering threshold

Again we see that filtering on rule length is very successful. See section 6.4.3 for further comments.

6.5.4 Filtering on reduct support

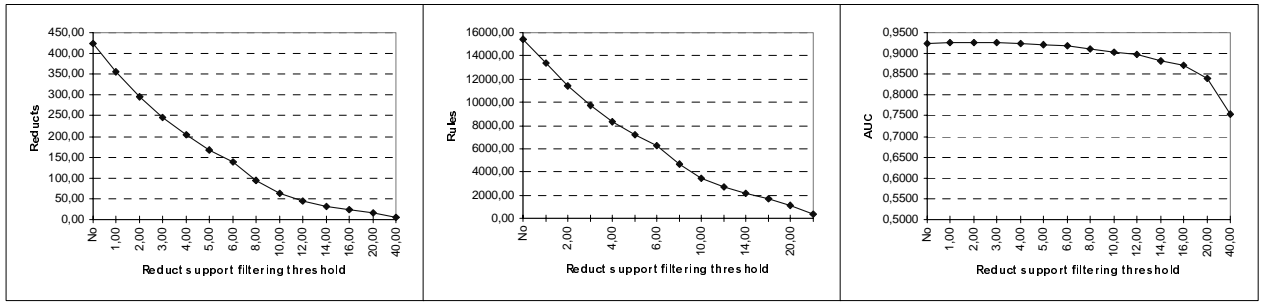


Figure 41: Filtering on reduct support; Reducts, rules and AUC as a function of filtering threshold

The AUC is pretty stable up to a threshold of 6. At this threshold, the reduct set is of size 138.25 and rule base consists of 6290.9 rules. This rule base is too high. When more reducts (and thus rules) are filtered away, the decrease in the AUC is relatively large.

6.5.5 Filtering by removing single attributes

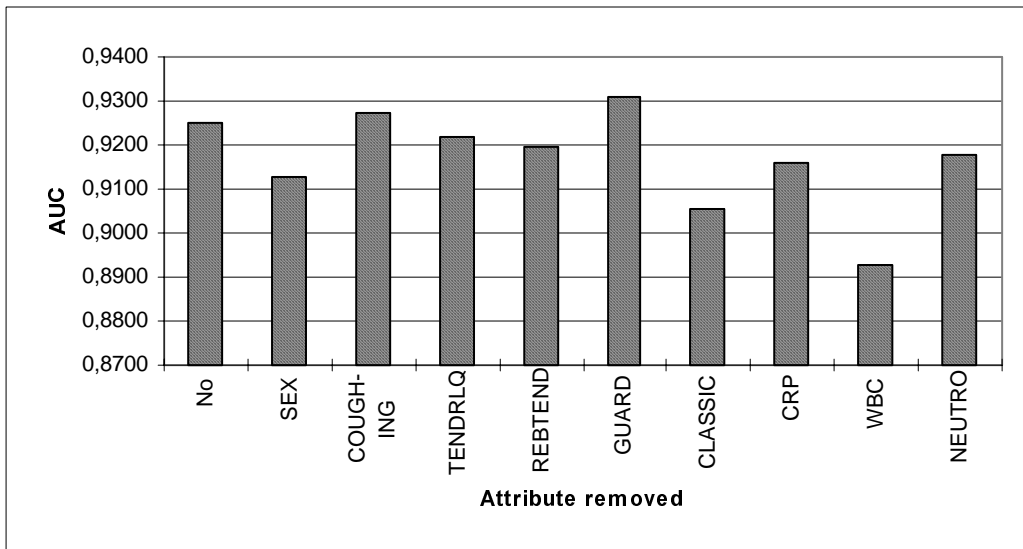


Figure 42: Filtering by removing single attributes; AUC as a function of filtering attribute

We see from the above figure that we get an increase in the AUC when reducts containing GUARD are removed from the reduct set. The models get an AUC of 0,9309 which is the highest achieved in this thesis. ALSO for COUGHING we get an increase in the AUC. All other attribute filterings imply a loss in the AUC. The attributes may be ranked as follows (in decreasing order): WBC, CLASSIC, SEX, CRP, NEUTRO, REBTEND, TENDRLQ, COUGHING, and GUARD.

6.6 Experiment 6

Here, we are going to study the best result achieved. The best result was achieved in the preceding section by removing all reducts containing the attribute GUARD from a model built by means of an exhaustive calculation of full dynamic reducts with samplings 5%, 15%, 25%, 35%, 45%, 55%, 65%, 75%, 85%, 95% of the size of the original data table. On each level reducts were searched for 10 times on different samples. A ROC curve for a model representative for the best result (the average of 20 models) is shown in Figure 43 The AUC is 0.9294.

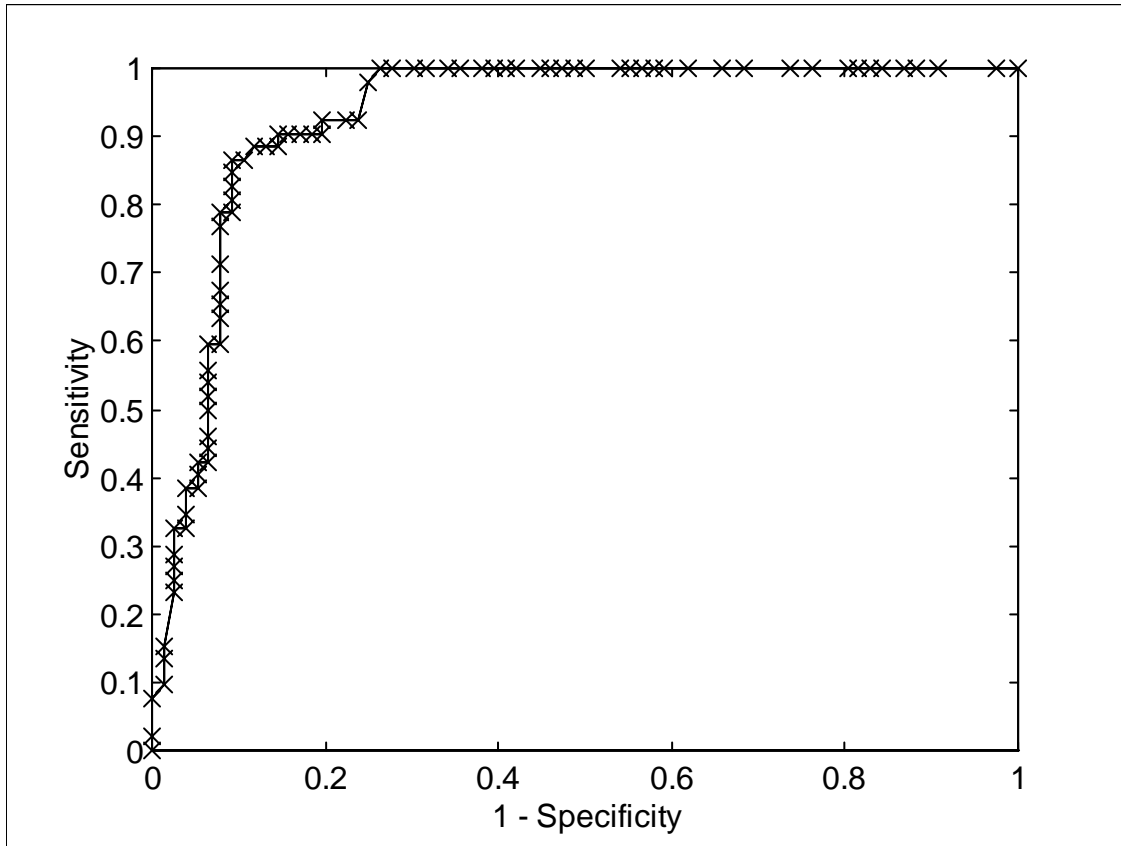


Figure 43: ROC curve for a representative of the best result

In the following curve the accuracy as a function of the prioritization threshold (for the negative diagnosis) is plotted for the same representative. We see that the top point is for a lower prioritization threshold than the typical threshold of 0.5. The best threshold is for a prioritization threshold of 0.428. The accuracy is for this threshold 0.8906, while the threshold of 0.5 it is only 0.8125. This is a typical example of how important the choice of the decision threshold is.

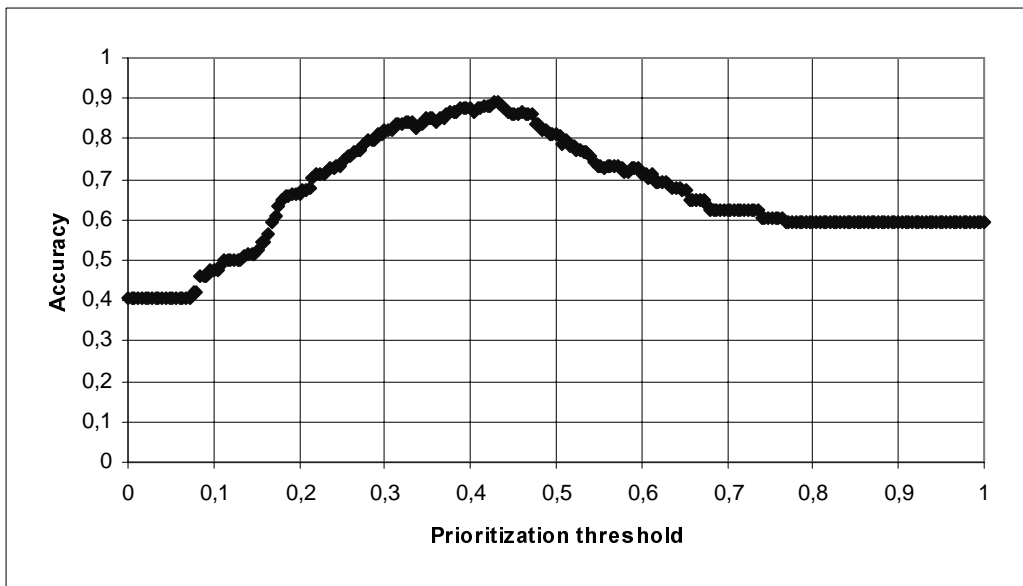


Figure 44: Accuracy as a function the prioritization threshold

We can see from the figure that the model is biased in the direction of the negative diagnosis. This is because the prevalence of disease is lower than 0.5 in the data table, and the rough set models generates more rules that support the majority decision, which is the negative diagnosis.

6.7 Experiment 7

In this section filtering on the reduct performance is done. The experiment was done as follows. The original appendicitis data set was split into 3 sets of equal size: One set is used for training, one set («tuning set») is the reducts' performance evaluated on, and the last is the testing set. As this process can not be automated in Rosetta, only one such splitting was done, because of lack of time.

The actual splitting had the following means for the decision attribute DIAG: Training set had a mean of 0.3488, the tune set had 0.407, and the test set had 0.3882.

Reducts were computed by means of full reducts using the genetic algorithm that was set to search for up to 10 reducts for each subsample in the dynamic reduct computation process. The sampling strategy was to sample 50 subtables of sizes 10%, 30%, 50%, 70%, and 90% of the original table size (10 samples on each level).

With no filtering, there were 469 reducts. The reducts were then ranked according to how well they performed on accuracy on the tuning set. Reducts that scored lower than the filtering threshold was removed. Then rules were generated from the remaining reducts and the training set, and the performance on accuracy, sensitivity, specificity, and AUC was recorded when the rule set was applied to the test set. The accuracy, sensitivity, and specificity were recorded on a majority voting as described in section 3.3.1.

The results of the experiment are shown in the following table.

Filtering threshold	# reducts	# rules	Accuracy	Sensitivity	Specificity	AUC
no filtering	469	20774	0,8235	0,6667	0,9231	0,9464
0,55	462	20626	0,8235	0,6667	0,9231	0,9467
0,6	418	19178	0,8588	0,7576	0,9231	0,9476
0,65	315	14375	0,8588	0,7576	0,9231	0,9519
0,7	109	4527	0,8824	0,8485	0,9038	0,9551
0,75	32	1107	0,8588	0,7879	0,9038	0,9523
0,8	4	119	0,8471	0,8485	0,8462	0,9382
0,83	1	30	0,8235	0,7273	0,8846	0,8097

Table 13: Results from filtering on reduct performance

We see from the table that both accuracy and AUC rise to a maximum value for a filtering threshold of 0.7. The AUC increases only from 0.9464 to 0.9519 while the accuracy increases considerably from 0.8235 to 0.8824. The reason for the this high increase can be seen on the sensitivity and specificity. There is a shift from a conservative model to a more risky model, as the sensitivity increases and the specificity decreases with increasing amounts of filtering.

The result in Table 13 shows that filtering on reduct performance might be useful. More experiments should be run to confirm this, however. The method has also been used in [VOF98].

7. Discussion and Conclusion

The analysis of a database, as is the case in this analysis, is very static compared to a doctor's way of making a diagnosis. For example, vomiting is preceded by anorexia and nausea in most instances except in diseases of the central nervous system. The appendicitis database is a very static snap-shot of the situation at one point in time. A doctor having a patient with abdominal pain may take advantage of the time, and has the opportunity to place a patient to observation in the hospital with for example frequent blood-tests taken from the patient. One additional point is that the patients may be in different stages of the development of for example acute appendicitis.

A very interesting project would be to test how well the best computer models perform on a similar set of patients in another hospital. This is actually done on a medical domain in [VOF98]. There is a risk that the rough set models are overoptimistic. As they are a result of iterations and backtrackings with optimisations of parametres, statistically invalid conclusions *may* have been reached [Sal97].

Logistic regression has the advantage of being a well-known and extensively used statistical method, based on rigorous mathematics. One disadvantage is that it is a linear method. The rough set approach, on the other hand is non-linear.

RS strengths: explainable rules, non-linear

RS weaknesses: discretization is a problem, problems with time dependencies

LR strengths: based on well-known and rigorous statistics

LR weaknesses: linearity, problems with time dependencies

The great advantage with the rough set approach is the explainability of the rules. The rules used in the classification can be read and immediately understood by humans. When finished with a rough set classification task, all generated rules (or only the most interesting) should be interpreted by medical experts. New medical insight may then be revealed. The rules should also be tested on independently collected data sets.

In order to get a better assessment of the relative "goodness" of the two models, their predictive capability should be assessed on a new independent set of data. There is a risk that both models may be over-optimistic.

Using the rough set approach we can support the conclusion in [HÅE97b] that biochemical tests seem to improve the computer models, and should, if used rationally, also provide physicians with important information in the investigation of acute appendicitis.

The rough set approach has in this thesis shown to perform very well on a data set that logistic regression model performed well on. With the additional advantage of the explainability of the rough set rules, the rough set models can be said to be better than the logistic regression models. One drawback is the high number of rules in the best performing rough set models. Nevertheless, it is possible to point out certain rules that performed particularly well. These rules should be validated by medical doctors, and in addition tested on data that have been collected independently of the database studied in this thesis.

Even though there are many possibilities for reducing a large rule set, using different filtering strategies, the most promising filtering method was actually to remove attributes from the analysis. The rough set approach gets confused when many (unimportant) attributes take part in the analysis. Thus, one important task when mining databases with the rough set approach is to identify the «best» attributes, and exclude the less informative from the analysis. Still, this is a difficult task.

The area under the ROC curve should be the preferred performance measure when generating rough set classifiers in ROSETTA with binary outcome.

Models that give predictions of each particular object in the form of a continuous score or prognostic index (as for instance logistic regression models and rough set models) is well suited for ROC analysis. Measuring the performance using ROC analysis and especially the area under the ROC curve (AUC) computed with the composite trapezoid rule seems to have some evident advantages to frequently used measures in the data mining and machine learning fields as error rate or accuracy. The AUC is independent of the prevalence in the data table, as it is built from pairs of sensitivity and specificity. Because of this, models built from samples with different

prevalences may be compared using the AUC. The AUC is also independent of any particular choice of a decision threshold, as all thresholds are taken into account in the generation of the ROC curve. By doing this, the typical problem of choosing a decision threshold is avoided. Last, but not least, the intuitive meaning of the AUC as the probability of a correct ranking of a (healthy, sick) pair of patients is an intuitive feature of a model to be maximized. The distribution of points in the plot should be analyzed carefully, however. One should plot the ROC curves and check that they look «normal»..

Because of this, I recommend that the AUC is used for measuring the performance of models with outcomes in the form of a continuous score or prognostic index. In particular, an AUC analysis seems to be well suited for assessing the performance of rough set models, and should therefore be a preferred performance measure when building such models. An important restriction is of course that the outcome variable (decision attribute) is binary.

8. Bibliography

- [Alt91] D. G. Altman (1991), *Practical Statistics for Medical Research*, London: Chapman and Hall.
- [BSS94] J. Bazan, A. Skowron, P. Synak (1994), Dynamic Reducts as a Tool for Extracting Laws from Decision Tables, Proc. Symp. on Methodologies for Intelligent Systems, Charlotte, NC, USA, October 16-19, Lecture Notes in Artificial Intelligence, Springer Verlag, Vol. 869, pp. 346-355.
- [Baz98] J. Bazan (1998), A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables, To appear in *Rough Sets in Knowledge Discovery*, L. Polkowski and A. Skowron (eds.), Physica Verlag.
- [CKØ98] U. Carlin, J. Komorowski, A. Øhrn (1998), Rough Set Analysis of Patients With Suspected Acute Appendicitis, Accepted for The Seventh Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, July 6 - 10, Université de La Sorbonne, Paris, France.
- [Eri96] S. Eriksson (1996), Acute Appendicitis - Ways to Improve Diagnostic Accuracy, *European Journal of Surgery*, Vol. 162, pp. 435-442.
- [FPSS96a] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth (1996), From Data Mining to Knowledge Discovery: An Overview, In U. Fayyad et al. (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Menlo Park, CA, pp. 1-34.
- [FPSS96b] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth (1996), From Data Mining to Knowledge Discovery in Databases, *AI Magazine*, Vol. 17, No. 3, pp. 37-54.
- [GFI94] J. M. Grönroos, J. J. Forsström, K. Irjala, et al. (1994), Phospholipase A₂, C-Reactive Protein, and White Blood Cell Count in the Diagnosis of Acute Appendicitis, *Clinical Chemistry*, 40, pp. 1757-1760.
- [HM82] J. A. Hanley, B. J. McNeil (1982), The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve, *Radiology*, 143, 29-36.
- [HÅE97a] S. Hallan, A. Åsberg, T.-H. Edna (1997), Estimating the Probability of Acute Appendicitis Using Clinical Criteria of a Structured Record Sheet: The Physician Against the Computer, *European Journal of Surgery*, Vol. 163, No 6, pp. 427-432.
- [HÅE97b] S. Hallan, A. Åsberg, T.-H. Edna (1997), Additional Value of Biochemical Tests in Suspected Acute Appendicitis, *European Journal of Surgery*, Vol. 163, No 7, pp. 533-538.
- [OZT93] W. P. Oosterhuis, A. H. Zwinderman, M. Teeuwen, et al. (1993), C reactive protein in the diagnosis of acute appendicitis, *European Journal of Surgery*, 159, pp. 115-119.
- [Paw82] Z. Pawlak (1982), Rough Sets, *International Journal of Computer and Information Sciences*, Vol. 11, No 5, pp. 341-356.
- [Paw91] Z. Pawlak (1991), *Rough Sets - Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht.
- [Sal97] S. L. Salzberg (1997), On comparing Classifiers: Pitfalls to Avoid and a Recommended Approach, In Usama Fayyad et al. (eds.), *Data Mining and Knowledge Discovery*, 1, Kluwer Academic Publishers, pp 317-328.
- [Sko93] A. Skowron (1993), Boolean Reasoning for Decision Rules Generation, In J. Komorowski and Z. W. Rás (eds.), *Seventh International Symposium for Methodologies for Intelligent Systems, ISMIS '93*, Trondheim, Norway, June 1993, Springer Verlag, pp. 295-305.

[Sko95] A. Skowron (1995), Synthesis of Adaptive Decision Systems from Experimental Data, Proc. Fifth Scandinavian Conference on Artificial Intelligence, Trondheim, Norway, May 29-31, *Frontiers in Artificial Intelligence and Applications*, A. Aamodt and J. Komorowski (eds.), IOS Press, Vol. 28, pp. 220-238.

[SR92] A. Skowron and C. Rauszer, The Discernibility Matrices and Functions in Information Systems, In R. Slowinski (ed.), *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory*, Dordrecht, Kluwer, 1992, pp 331-362.

[Slo88] K. Slowinski (1988), Rough Set Approach to Analysis of Data from Peritoneal Lavage in Acute Pancreatitis, *Medical Informatics*, 13, no. 3, pp 143-159.

[Slo92] K. Slowinski (1992), Rough classification of HSV patients, In R. Slowinski (ed.), *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, pp. 77-94.

[Swe88] J. A. Swets (1988), Measuring the Accuracy of Diagnostic Systems, *Science*, Vol. 240, 1285-1293.

[VOF98] S. Vinterbo, L. Ohno-Machado, H. Fraser (1998), Prediction of Acute Myocardial Infarction using Rough Sets, In preparation.

[Wro95] J. Wroblewski (1995), Finding Minimal Reducts using Genetic Algorithms (Extended Version), Proc. Second International Joint Conference on Information Sciences, Wrightsville Beach, NC, USA, September 28-October 1, pp. 186-189.

[ZC93] M. H. Zweig, G. Campbell (1993), Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine, *Clinical Chemistry*, 39, 561-577.

[Øhrn] A. Øhrn, A Unified Exposition of Some Reduct Types, The Norwegian University of Science and Technology, In preparation.

[ØK97] A. Øhrn, J. Komorowski (1997), ROSETTA - A Rough Set Toolkit for Analysis of Data, Proc. Third International Joint Conference on Information Sciences, Durham, NC, USA, March 1-5, Vol. 3, pp 403-407.

[ØKSS98] A. Øhrn, J. Komorowski, A. Skowron, P. Synak (1998), The Design and Implementation of a Knowledge Discovery Toolkit Based on Rough Sets - The ROSETTA System, To appear in *Rough Sets in Knowledge Discovery*, L. Polkowski and A. Skowron (eds.), Physica Verlag, 24 pages.

[ØVSK97] A. Øhrn, S. Vinterbo, P. Szymanski, J. Komorowski (1997), Modelling Cardiac Patient Set Residuals using Rough Sets, Proc. AMIA Annual Fall Symposium (formerly SCAMC), Nashville, TN, USA, October 25-29, pp. 203-207.

<http://www.twocrows.com/whitep.htm>

<http://www.human.cornell.edu/News/HNN31.html>

APPENDIX

Forsøk	Coverage	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yaa	0,02		628,80				0,9114	0,0274
b	0,04		597,15				0,9113	0,0275
c	0,06		547,45				0,9114	0,0274
d	0,08		499,50				0,9113	0,0275
e	0,10		440,65				0,9114	0,0277
f	0,12		403,65				0,9119	0,0279
g	0,14		345,15				0,9120	0,0281
h	0,16		308,40				0,9121	0,0283
i	0,18		254,70				0,9122	0,0288
j	0,20		215,45				0,9128	0,0285
k	0,22		190,25				0,9116	0,0285
l	0,24		163,50				0,9119	0,0285
m	0,26		140,55				0,9103	0,0290
n	0,28		119,00				0,9088	0,0283
o	0,30		101,35				0,9066	0,0293
p	0,32		90,05				0,9047	0,0293
q	0,34		74,40				0,9014	0,0280
r	0,36		60,45				0,8984	0,0297
s	0,38		51,25				0,8946	0,0298
t	0,40		42,00				0,8932	0,0293
u	0,42		36,05				0,8918	0,0295
v	0,44		30,60				0,8873	0,0269
w	0,46		25,40				0,8828	0,0311
x	0,48		21,95				0,8814	0,0281
y	0,50		17,65				0,8765	0,0273
z	0,52		16,10				0,8734	0,0238
1	0,54		13,05				0,8557	0,0595
2	0,56		11,15				0,8320	0,1081
3	0,58		9,05				0,8177	0,1045
4	0,60		6,95				0,7730	0,1315
5	0,62		5,50				0,7504	0,1243
6	0,64		4,35				0,7131	0,1285
7	0,66		3,45				0,6915	0,1160
8	0,68		2,75				0,6709	0,1170
9	0,70		2,50				0,6498	0,1112
Forsøk	Probability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yba	0,50		593,75				0,9117	0,0274
b	0,60		491,95				0,9107	0,0279
c	0,70		376,20				0,9073	0,0290
d	0,80		227,95				0,8998	0,0312
e	0,90		127,85				0,8647	0,0337
f	0,99		73,65				0,7997	0,0475
Forsøk	Rule stability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yca	0,10		358,15				0,9050	0,0280
b	0,20		262,65				0,9017	0,0269
c	0,30		169,00				0,8893	0,0282
d	0,40		151,30				0,8848	0,0277
e	0,50		61,70				0,7852	0,0502
f	0,60		61,10				0,7852	0,0504

g	0,70		58,70				0,7783	0,0544
h	0,80		46,75				0,7473	0,0516
i	0,90		32,50				0,6831	0,0607
Forsøk	Reduct length	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yda	5	314,70	632,10				0,9113	0,0274
b	4	314,00	631,40				0,9113	0,0275
c	3	163,80	456,20				0,9106	0,0280
d	2	18,50	31,20				0,8969	0,0292
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yea	1	175,45	492,20				0,9106	0,0270
b	2	134,90	425,40				0,9093	0,0272
c	3	113,25	381,90				0,9072	0,0285
d	4	96,90	346,35				0,9041	0,0295
e	5	82,15	309,65				0,8991	0,0310
f	6	69,45	275,80				0,8950	0,0317
g	8	49,45	214,65				0,8885	0,0311
h	10	35,60	166,00				0,8783	0,0325
i	12	25,35	124,85				0,8752	0,0291
j	14	17,20	86,60				0,8613	0,0355
k	16	11,75	61,30				0,8494	0,0365
l	20	5,80	28,10				0,8235	0,0743
m	40	1,25	1,75				0,5762	0,1359
Forsøk	Coverage	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
abc1u5	No	512,25	35583,50	0,7144	0,8867	0,8160	0,9113	0,0295
zaa	0,02		17898,35				0,9110	0,0295
b	0,04		5116,15				0,9098	0,0290
c	0,06		2370,05				0,9090	0,0297
d	0,08		1315,55				0,9080	0,0301
e	0,10		793,35				0,9065	0,0307
f	0,12		592,50				0,9058	0,0302
g	0,14		410,40				0,9043	0,0301
h	0,16		322,30				0,9034	0,0301
i	0,18		231,60				0,9007	0,0298
j	0,20		175,45				0,8990	0,0303
k	0,22		144,00				0,8984	0,0284
l	0,24		115,45				0,8976	0,0282
m	0,26		93,25				0,8937	0,0276
n	0,28		74,60				0,8903	0,0263
o	0,30		59,05				0,8834	0,0247
p	0,32		50,40				0,8804	0,0269
q	0,34		40,85				0,8759	0,0257
r	0,36		32,80				0,8726	0,0285
s	0,38		27,95				0,8664	0,0298
t	0,40		22,05				0,8523	0,0396
u	0,42		19,50				0,8472	0,0450
v	0,44		16,80				0,8359	0,0481
w	0,46		14,20				0,8303	0,0444
x	0,48		12,05				0,8183	0,0580
y	0,50		9,75				0,8152	0,0575
z	0,52		9,55				0,8142	0,0578
1	0,54		8,05				0,7903	0,0648
2	0,56		6,95				0,7736	0,0637
3	0,58		5,90				0,7665	0,0640
Forsøk	Probability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD

abclu5	No	512,25	35583,50	0,7144	0,8867	0,8160	0,9113	0,0295
zba	0,50		33989,20				0,9114	0,0295
b	0,60		33687,25				0,9110	0,0291
c	0,70		32685,75				0,9106	0,0283
d	0,80		31989,60				0,9084	0,0280
e	0,90		31737,05				0,9053	0,0242
f	0,99		31670,70				0,9020	0,0237
Forsøk	Reduct length	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
abclu5	No	512,25	35583,50	0,7144	0,8867	0,8160	0,9113	0,0295
f	7	391,70	22568,85				0,9110	0,0297
e	6	266,20	11110,35				0,9104	0,0298
zda	5	155,20	3681,00				0,9075	0,0309
b	4	93,55	1210,30				0,9042	0,0324
c	3	30,85	199,00				0,8915	0,0306
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
abclu5	No	512,25	35583,50	0,7144	0,8867	0,8160	0,9113	0,0295
zea	1	58,10	4692,95				0,8886	0,0270
b	2	16,00	1513,40				0,8311	0,0518
c	3	6,55	668,05				0,7514	0,0549
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
r1a	No	314,70	632,10	0,5319	0,9291	0,7717	0,9113	0,0274
yfa	AGE	262,45	522,85				0,9118	0,0278
b	SEX	273,00	566,30				0,8979	0,0283
c	DURATION	261,75	521,35				0,9145	0,0269
d	ANOREXIA	282,75	583,75				0,9116	0,0276
e	NAUSEA	284,00	583,65				0,9121	0,0276
f	PREVIOUS	299,20	608,65				0,9117	0,0275
g	MOVEMENT	276,25	575,90				0,9114	0,0289
h	COUGHING	278,30	576,60				0,9102	0,0283
i	MICTUR	289,15	590,40				0,9115	0,0282
j	TENDRLQ	307,30	624,65				0,9108	0,0274
k	REBTEND	271,35	562,70				0,9010	0,0309
l	GUARD	275,10	568,70				0,9102	0,0270
m	CLASSIC	268,70	554,95				0,8884	0,0280
n	TEMP	258,60	515,65				0,9106	0,0284
o	ESR	268,65	535,70				0,9133	0,0270
p	CRP	263,40	515,70				0,9028	0,0263
q	WBC	255,25	500,25				0,8821	0,0310
r	NEUTRO	260,90	514,75				0,9113	0,0273
s	LEFT	280,70	577,80				0,9109	0,0263
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	SD
abclu5	No	512,25	35583,50	0,7144	0,8867	0,8160	0,9113	0,0295
zfa	AGE	247,15	13198,85				0,9071	0,0288
b	SEX	363,25	23548,30				0,9011	0,0293
c	DURATION	309,00	18724,80				0,9140	0,0296
d	ANOREXIA	448,25	30763,35				0,9111	0,0298
e	NAUSEA	445,05	30533,50				0,9125	0,0298
f	PREVIOUS	504,95	35149,10				0,9115	0,0293
g	MOVEMENT	411,20	27549,90				0,9098	0,0297
h	COUGHING	423,50	28672,35				0,9109	0,0302
i	MICTUR	470,90	32553,30				0,9104	0,0294
j	TENDRLQ	491,75	34274,50				0,9106	0,0296
k	REBTEND	377,90	25348,30				0,9020	0,0308
l	GUARD	381,50	24857,75				0,9095	0,0289
m	CLASSIC	339,25	22302,65				0,8888	0,0299

n	TEMP	299,80	17948,60				0,9127	0,0308
o	ESR	338,10	21202,45				0,9119	0,0296
p	CRP	292,00	18620,15				0,9009	0,0327
q	WBC	191,05	10382,90				0,8744	0,0353
r	NEUTRO	290,15	17535,50				0,9116	0,0286
s	LEFT	450,30	31357,65				0,9115	0,0301

Forsøk	Coverage	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD
abc\lx3h1	No	54,35	522,25	0,5852	0,8769	0,7600	0,8495 0,0263
x3h1aa	0,02		471,50				0,8493 0,0263
ab	0,04		407,35				0,8488 0,0260
ac	0,06		354,35				0,8488 0,0262
ad	0,08		305,05				0,8486 0,0265
ae	0,10		255,55				0,8485 0,0266
af	0,12		228,10				0,8480 0,0268
ag	0,14		197,00				0,8469 0,0265
ah	0,16		176,20				0,8463 0,0265
ai	0,18		152,60				0,8462 0,0265
aj	0,20		134,15				0,8458 0,0274
ak	0,22		120,75				0,8464 0,0264
al	0,24		108,40				0,8457 0,0267
am	0,26		95,75				0,8445 0,0260
an	0,28		84,20				0,8437 0,0264
ao	0,30		75,50				0,8423 0,0259
ap	0,32		69,70				0,8422 0,0254
aq	0,34		61,05				0,8412 0,0250
ar	0,36		53,40				0,8417 0,0249
as	0,38		48,60				0,8424 0,0248
at	0,40		42,35				0,8446 0,0249
au	0,42		38,80				0,8448 0,0242
av	0,44		35,15				0,8463 0,0250
aw	0,46		31,10				0,8473 0,0264
ax	0,48		28,60				0,8477 0,0270
ay	0,50		24,40				0,8502 0,0261
az	0,52		23,15				0,8487 0,0275
a1	0,54		20,30				0,8486 0,0288
a2	0,56		17,85				0,8455 0,0245
a3	0,58		16,45				0,8462 0,0224
a4	0,60		14,30				0,8353 0,0220
a5	0,62		12,20				0,8330 0,0240
a6	0,64		10,90				0,8255 0,0299
a7	0,66		9,40				0,8190 0,0262
a8	0,68		8,40				0,8094 0,0280
a9	0,70		8,05				0,7991 0,0269

Forsøk	Probability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD
abc\lx3h1	No	54,35	522,25	0,5852	0,8769	0,7600	0,8495 0,0263
x3h1ba	0,50		499,55				0,8495 0,0259
bb	0,60		441,30				0,8475 0,0256
bc	0,70		365,85				0,8391 0,0263
bd	0,80		270,55				0,6742 0,1129
be	0,90		213,35				0,4802 0,0867
bf	0,99		187,15				0,5942 0,0317

Forsøk	Reduct length	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD
abc\lx3h1	No	54,35	522,25	0,5852	0,8769	0,7600	0,8495 0,0263
df	7,00	54,35	522,25				0,8495 0,0263
de	6,00	53,40	489,45				0,8495 0,0262

x3h1da	5,00	47,70	359,95				0,8495	0,0261
db	4,00	36,30	199,70				0,8483	0,0267
dc	3,00	18,80	64,75				0,8464	0,0253
dd	2,00	5,20	10,40				0,8420	0,0275
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h1	No	54,35	522,25	0,5852	0,8769	0,7600	0,8495	0,0263
x3h1ea	1,00	44,45	430,75				0,8460	0,0268
eb	2,00	34,90	351,00				0,8422	0,0273
ec	3,00	28,00	289,55				0,8396	0,0248
ed	4,00	22,55	243,70				0,8391	0,0239
ee	5,00	16,40	194,75				0,8328	0,0263
ef	6,00	12,50	162,40				0,8258	0,0305
eg	8,00	7,00	117,60				0,8211	0,0303
eh	10,00	4,50	93,40				0,8164	0,0203
ei	12,00	3,50	82,65				0,8091	0,0206
ej	14,00	2,60	70,40				0,8020	0,0259
ek	16,00	2,30	65,70				0,7985	0,0270
el	20,00	2,00	60,95				0,7980	0,0258
em	40,00	1,95	60,15				0,7945	0,0270
Forsøk	Coverage	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h2	No	105,15	1744,50	0,7215	0,8896	0,8210	0,9061	0,0250
x3h2aa	0,02		1471,90				0,9060	0,0251
ab	0,04		1082,90				0,9052	0,0254
ac	0,06		842,30				0,9050	0,0255
ad	0,08		661,30				0,9035	0,0258
ae	0,10		515,15				0,9034	0,0257
af	0,12		436,25				0,9022	0,0258
ag	0,14		350,75				0,9013	0,0257
ah	0,16		304,05				0,8998	0,0250
ai	0,18		248,90				0,8974	0,0249
aj	0,20		212,05				0,8955	0,0246
ak	0,22		185,65				0,8940	0,0244
al	0,24		159,35				0,8917	0,0242
am	0,26		137,15				0,8895	0,0231
an	0,28		115,85				0,8886	0,0223
ao	0,30		100,50				0,8858	0,0228
ap	0,32		91,35				0,8859	0,0231
aq	0,34		78,75				0,8845	0,0229
ar	0,36		67,35				0,8847	0,0229
as	0,38		60,85				0,8855	0,0231
at	0,40		51,50				0,8850	0,0221
au	0,42		46,75				0,8835	0,0212
av	0,44		41,90				0,8819	0,0223
aw	0,46		36,25				0,8809	0,0230
ax	0,48		32,45				0,8800	0,0243
ay	0,50		27,35				0,8787	0,0263
az	0,52		26,00				0,8785	0,0276
a1	0,54		22,50				0,8710	0,0256
a2	0,56		19,65				0,8659	0,0248
a3	0,58		17,75				0,8588	0,0249
a4	0,60		14,95				0,8422	0,0257
a5	0,62		12,55				0,8357	0,0265
a6	0,64		10,95				0,8265	0,0278
a7	0,66		9,45				0,8204	0,0243
a8	0,68		8,25				0,8093	0,0282
a9	0,70		7,90				0,7990	0,0271

Forsøk	Probability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h2	No	105,15	1744,50	0,7215	0,8896	0,8210	0,9061	0,0250
x3h2ba	0,50		1636,40				0,9058	0,0250
bb	0,60		1524,25				0,9067	0,0267
bc	0,70		1344,75				0,9086	0,0288
bd	0,80		1120,10				0,8881	0,0344
be	0,90		973,30				0,7466	0,0832
bf	0,99		913,45				0,7030	0,0508
Forsøk	Reduct length	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h2	No	105,15	1744,50	0,7215	0,8896	0,8210	0,9061	0,0250
df	7,00	104,55	1705,45				0,9060	0,0251
de	6,00	98,60	1419,15				0,9059	0,0254
x3h2da	5,00	82,10	888,95				0,9051	0,0256
db	4,00	55,50	386,25				0,9035	0,0256
dc	3,00	25,55	102,55				0,9030	0,0271
dd	2,00	6,20	13,40				0,8966	0,0268
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h2	No	105,15	1744,50	0,7215	0,8896	0,8210	0,9061	0,0250
x3h2ea	1,00	84,85	1445,95				0,9057	0,0267
eb	2,00	68,35	1216,10				0,9069	0,0266
ec	3,00	55,05	1037,20				0,9080	0,0254
ed	4,00	44,65	881,40				0,9086	0,0268
ee	5,00	35,00	729,95				0,9079	0,0285
ef	6,00	28,25	629,20				0,9037	0,0299
eg	8,00	18,40	488,20				0,8999	0,0292
eh	10,00	12,75	383,85				0,8921	0,0324
ei	12,00	9,65	322,25				0,8846	0,0319
ej	14,00	6,90	258,60				0,8746	0,0297
ek	16,00	5,50	229,15				0,8608	0,0383
el	20,00	3,85	181,95				0,8390	0,0318
em	40,00	1,95	111,25				0,8078	0,0381
Forsøk	Coverage	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h3	No	423,50	15483,85	0,7517	0,9030	0,8413	0,9249	0,0245
x3h3aa	0,02		10646,70				0,9249	0,0245
ab	0,04		5513,40				0,9250	0,0244
ac	0,06		3441,40				0,9246	0,0242
ad	0,08		2260,90				0,9236	0,0241
ae	0,10		1496,55				0,9225	0,0241
af	0,12		1157,50				0,9219	0,0237
ag	0,14		829,20				0,9206	0,0238
ah	0,16		672,90				0,9194	0,0236
ai	0,18		500,45				0,9174	0,0241
aj	0,20		396,65				0,9159	0,0242
ak	0,22		336,05				0,9149	0,0239
al	0,24		273,40				0,9133	0,0239
am	0,26		226,25				0,9107	0,0238
an	0,28		181,60				0,9090	0,0246
ao	0,30		149,70				0,9062	0,0247
ap	0,32		131,00				0,9054	0,0259
aq	0,34		107,65				0,9022	0,0250
ar	0,36		88,55				0,9012	0,0256
as	0,38		77,30				0,9000	0,0259
at	0,40		62,20				0,8978	0,0253
au	0,42		55,60				0,8969	0,0243
av	0,44		48,00				0,8934	0,0258
aw	0,46		40,30				0,8928	0,0267

ax	0,48		35,40				0,8899	0,0266
ay	0,50		29,40				0,8878	0,0284
az	0,52		27,35				0,8840	0,0282
a1	0,54		23,20				0,8736	0,0267
a2	0,56		20,20				0,8678	0,0254
a3	0,58		17,95				0,8582	0,0246
a4	0,60		14,95				0,8425	0,0260
a5	0,62		12,50				0,8358	0,0267
a6	0,64		10,90				0,8267	0,0280
a7	0,66		9,40				0,8205	0,0247
a8	0,68		8,20				0,8098	0,0290
a9	0,70		7,85				0,7995	0,0281
Forsøk	Probability	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h3	No	423,50	15483,85	0,7517	0,9030	0,8413	0,9249	0,0245
x3h3ba	0,50		14376,05				0,9247	0,0247
bb	0,60		13944,50				0,9250	0,0246
bc	0,70		12888,45				0,9254	0,0254
bd	0,80		11889,95				0,9213	0,0263
be	0,90		11408,65				0,9020	0,0207
bf	0,99		11257,35				0,8840	0,0203
Forsøk	Reduct length	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h3	No	423,50	15483,85	0,7517	0,9030	0,8413	0,9249	0,0245
df	7,00	395,70	13046,95				0,9250	0,0246
de	6,00	328,65	8610,05				0,9252	0,0244
x3h3da	5,00	222,20	3847,40				0,9253	0,0244
db	4,00	117,70	1138,60				0,9247	0,0247
dc	3,00	42,40	206,85				0,9233	0,0255
dd	2,00	8,20	19,40				0,9195	0,0257
Forsøk	Reduct support	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h3	No	423,50	15483,85	0,7517	0,9030	0,8413	0,9249	0,0245
x3h3ea	1,00	354,90	13336,00				0,9256	0,0249
eb	2,00	296,10	11431,15				0,9256	0,0250
ec	3,00	246,40	9766,55				0,9255	0,0253
ed	4,00	203,35	8368,65				0,9239	0,0255
ee	5,00	167,65	7240,85				0,9214	0,0267
ef	6,00	138,25	6290,90				0,9175	0,0270
eg	8,00	93,00	4641,85				0,9115	0,0256
eh	10,00	63,50	3473,50				0,9029	0,0267
ei	12,00	44,85	2683,35				0,8976	0,0282
ej	14,00	32,50	2105,50				0,8811	0,0272
ek	16,00	24,30	1715,40				0,8714	0,0277
el	20,00	14,45	1137,45				0,8401	0,0360
em	40,00	4,30	389,65				0,7536	0,0608
Forsøk	Attribute	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h1	No	54,35	522,25	0,5852	0,8769	0,7600	0,8495	0,0263
x3h1fa	SEX	24,50	159,20				0,8130	0,0306
b	COUGH-ING	26,35	179,45				0,8452	0,0240
c	TENDRLQ	30,70	232,90				0,8412	0,0274
d	REBTEND	26,70	184,40				0,8226	0,0245
e	GUARD	26,35	181,00				0,8485	0,0320
f	CLASSIC	26,80	187,30				0,8122	0,0246
Forsøk	Attribute	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC ± SD	
abc\lx3h2	No	105,15	1744,50	0,7215	0,8896	0,8210	0,9061	0,0250
x3h2fa	SEX	49,70	596,05				0,8907	0,0296
b	COUGH-ING	52,45	633,15				0,9085	0,0253

c	TENDRLQ	61,95	856,85				0,8997	0,0268
d	REBTEND	54,20	686,45				0,8969	0,0245
e	GUARD	52,70	659,00				0,9139	0,0279
f	CLASSIC	53,80	674,45				0,8841	0,0276
g	WBC	47,70	409,80				0,8492	0,0263
Forsøk	Attribute	Reducts	Rules	Sensitivity	Specificity	Accuracy	AUC	± SD
abc\x3h3	No	423,50	15483,85	0,7517	0,9030	0,8413	0,9249	0,0245
x3h3fa	SEX	210,35	6346,85				0,9129	0,0284
b	COUGH-ING	214,95	6424,55				0,9272	0,0248
c	TENDRLQ	250,85	8448,85				0,9217	0,0259
d	REBTEND	223,45	6809,35				0,9198	0,0246
e	GUARD	216,15	6559,40				0,9309	0,0235
f	CLASSIC	223,95	6927,15				0,9055	0,0267
g	CRP	207,35	5214,90				0,9157	0,0237
h	WBC	205,00	5424,65				0,8928	0,0261
i	NEUTRO	205,70	5272,25				0,9178	0,0246